



TUGAS AKHIR - TM 141586

**RANCANG BANGUN SISTEM PENGUKURAN
PERUBAHAN POSISI DENGAN MENGGUNAKAN
KAMERA TUNGGAL DAN LASER**

**IKHSAN ABI NUBLI
NRP 021113 4000 0137**

**DOSEN PEMBIMBING
ARIF WAHJUDI, ST., MT., PhD**

**LABORATORIUM PERANCANGAN DAN PENGEMBANGAN PRODUK
DEPARTEMEN TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018**



TUGAS AKHIR - TM 141586

**RANCANG BANGUN SISTEM PENGUKURAN
PERUBAHAN POSISI DENGAN MENGGUNAKAN
KAMERA TUNGGAL DAN LASER**

**IKHSAN ABI NUBLI
NRP 021113 4000 0137**

**DOSEN PEMBIMBING
ARIF WAJUDI, ST., MT., PhD**

**LABORATORIUM PERANCANGAN DAN PENGEMBANGAN PRODUK
DEPARTEMEN TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018**

(Halaman ini sengaja dikosongkan)



FINAL PROJECT - TM 141586

**DESIGN OF MEASUREMENT SHIFTING SYSTEM
USING MONO CAMERA AND LASER**

**IKHSAN ABI NUBLI
SN 021113 4000 0137**

**SUPERVISOR
ARIF WAHJUDI, ST., MT., PhD**

**PRODUCT DESIGN AND DEVELOPMENT LABORATORY
DEPARTMENT OF MECHANICAL ENGINEERING
FACULTY OF INDUSTRIAL TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018**

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM PENGUKURAN PERUBAHAN POSISI DENGAN MENGGUNAKAN KAMERA TUNGGAL DAN LASER

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
Pada
Program Studi S-1 Departemen Teknik Mesin
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :

IKHSAN ABI NUBLI

NRP. 021113 4000 0137

Disetujui oleh Tim Penguji Tugas Akhir

1. Arif Wahjudi, ST., MT. (Pembimbing)
NIP. 197303222001121081 (Penguji I)
2. Dinny Harnany, ST., MSc. (Pembimbing)
NIP. 210020 1405001 (Penguji II)
3. Ari Kurniawan Saputra, ST., MT.
NIP. 198604012015041001 (Penguji II)



**SURABAYA
JULI, 2018**

(Halaman ini sengaja dikosongkan)

RANCANG BANGUN SISTEM PENGUKURAN PERUBAHAN POSISI DENGAN MENGGUNAKAN KAMERA TUNGGAL DAN LASER

Nama Mahasiswa : Ikhwan Abi Nubli
NRP : 021113 4000 0137
Departemen : Teknik Mesin FTI-ITS
Dosen Pembimbing : Arif Wahjudi, ST., MT., PhD.

Abstrak

Pengukuran merupakan metode dalam menentukan nilai besaran dari suatu objek., image processing dapat digunakan untuk memperoleh data. Input yang berupa gambar atau video dapat diperoleh dengan menggunakan baik mono kamera maupun stereo kamera. Sudah ada penelitian untuk memanfaatkan metode image processing untuk mendapatkan koordinat, sehingga dari penelitian tersebut perlu adanya pengembangan lagi untuk pendekripsi jarak, kecepatan dan percepatan yang dideteksi dengan image processing.

Langkah penelitian pengukuran posisi ini secara garis besar terdiri dari tiga langkah, yaitu persiapan, perancangan dan pembuatan program serta proses analisa data. Langkah persiapan terdiri dari studi pustaka, lalu proses identifikasi masalah serta perumusan masalah, Setelah itu, dilakukan Pengambilan data jarak piksel dilakukan dengan berbagai macam jarak tertentu sehingga diperoleh data yang nanti dianalisis dengan metode regresi linear,. Jarak yang terdeteksi oleh program lalu dibagi dengan menggunakan fungsi waktu jalan program agar mendapatkan nilai kecepatan dan percepatan

Adapun hasil yang diperoleh antara lain, hasil pengukuran posisi dengan menggunakan laser dan mono-kamera, dengan rata – rata error yang dihasilkan adalah sebesar 1.365%, hasil pengukuran kecepatan dengan menggunakan laser dan mono-kamera dengan rata – rata error yang dihasilkan adalah sebesar 5.481%. dan diperoleh cara hasil pengukuran percepatan dengan

menggunakan laser dan mono-kamera dengan rata – rata error yang dihasilkan sebesar -0.0008%.

Kata Kunci : Pengukuran, Image Processing, Pythagoras, linear regresi, jarak piksel, kecepatan, percepatan, jarak, error

DESIGN OF MEASUREMENT SHIFTING SYSTEM USING MONO CAMERA AND LASER

Name	: Ikhsan Abi Nubli
Student Number	: 021113 4000 0137
Department	: Mechanical Engineering FTI-ITS
Supervisors	: Arif Wahjudi, ST., MT., PhD.

Abstract

Measuring is a method to scaling an object, image processing can be used for importing data of measurement. Frame which is consists of video or picture is used as an input while using mono camera or stereo camera. There's a research using image processing to get coordinates from a frame, so from that research, to improve until we could determine the position, velocity and acceleration of an object.

Steps for this position measurement, as we generalised to 3 steps which are: preparation, construction and designing the program and analyzing the data. Preparation steps consists of finding literature, problem identification and formulation of the problem. After that, retrieving pixel data by measuring with real distance until we get so much data that can be analyze using linear regression. The Distance that has been detected by program is divided by time lapse of the program to get the scale of velocity and acceleration

The result for distance measurement using mono camera and laser is with average error 1.365%, result for velocity measurement using mono camera and laser is with average error 5.481%. and result for acceleration measurement using mono camera and laser is with average error -0.0008%.

Keywords : Measurement, Image Processing, Pythagoras, linear regression, pixel, velocity, acceleration, distance, error

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan nikmat dan karunia Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “**RANCANG BANGUN SISTEM PENGUKURAN PERUBAHAN POSISI DENGAN MENGGUNAKAN KAMERA TUNGGAL DAN LASER**” dengan baik. Terselesaiannya Tugas Akhir ini tidak lepas dari bimbingan dan arahan dari berbagai pihak, sehingga pada kesempatan ini penulis mengucapkan terima kasih kepada :

1. Diri sendiri, karena apabila bukan karena diri sendiri, maka Tugas Akhir ini tidak dapat terselesaikan.
2. Fiqih Syamsu Rokhman, S.E, M.M dan Elmasa Wahyuni , S.E sebagai kedua orang tua, atas segala do'a, nasehat, kasih sayang, dan dukungan yang diberikan kepada penulis demi kesuksesan dan kebahagiaan penulis.
3. Bapak Arif Wahjudi, S.T, M.T, P.hD, selaku dosen pembimbing tugas akhir yang telah memberikan ilmu dan waktunya untuk menyelesaikan Tugas Akhir ini.
4. Ibu Dinny Harnany, ST., Msc, Bapak Ari Kurniawan Saputra, ST., MT, Prof. Dr. Ing. Ir. I Made Londen Batan M.Eng dan Unggul Wasiwitono ST., M.Eng, selaku dosen penguji yang telah memberikan kritik, saran serta masukan dalam penyelesaian Tugas Akhir ini.
5. Pak Sugeng, Pak Sokip dan Kepala Laboratorium Fisika Dasar Drs. Bachtra Indarto, M.Si yang telah memberikan fasilitas penunjang penelitian untuk penulis
6. Ariq Mahira dan Dzakia Audrey, adik – adik penulis yang sebagai penyemangat penulis untuk segera menyelesaikan tugas akhir ini.
7. Eka Aullya Risma Haqqi yang menjadi inspirasi dan semangat penulis menyelesaikan Tugas Akhir ini. “dengar jantung ini berdetak untukmu, berlututku meraih jari manismu, ku genggam tanganmu, *you know that I love you so much*”

8. Agung Wahyu, Masud Asadullah dan Hafizh Nazhar Pahlevi partner tugas akhir yang selama ini telah berjuang bersama dan saling memberikan semangat.
9. Teman – teman lab p3 terutama Tetew Club, (Akbar, Alwi, Kang Deni, Deris, Fakrul, Mas Hendi, Rado ST, Mario, Syamsul, Awan, Mbah Arif, Radian, Mas Chandra, Ayek, Ray , Rambo, Tedi, Tuba, Baskara, Heng dan Yunico) dan juga Rosa, Uzik dan Faiq yang selama ini telah membantu, mendukung, dan penulis selama masa mengerjakan tugas akhir di lab
10. Teman – Teman Jenderal Besar (hafizh, alwi, Culbot, adhit, Heng dan Imam) yang selama ini saling mendukung satu sama lain agar segera lulus kuliah
11. Mbak Nela dan Mas Ardi yang mengajari dasar OpenCV C++
12. Teman-teman Angkatan M56 dan Teknik Mesin ITS 2013 yang memberikan rasa kekeluargaan selama kuliah.
13. Pengurus Himpunan periode 2014-2015 dan periode 2015-2016 terutama Departemen KWU yang saling mendukung di kegiatan non-akademik
14. Squad New England Patriots musim 2015 dan musim 2017 atas *comeback*-nya yang luar biasa menjadi inspirasi penulis untuk percaya bahwa penulis bisa menyelesaikan studi S1 ini.
15. Semua pihak yang telah membantu dalam penulisan laporan ini, yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa Tugas Akhir masih jauh dari kesempurnaan. Sehingga saran dan kritik dari segala pihak yang bersifat membangun sangat penulis harapkan demi kesempurnaan penulisan selanjutnya.

Surabaya, Juli 2018

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
COVER PAGE	iii
LEMBAR PENGESAHAN.....	v
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
BAB II DASAR TEORI.....	3
2.1 Penelitian Sebelumnya	3
2.2 Kamera.....	4
2.3 Gambar Digital.....	5
2.4 Kedalaman Warna.....	5
2.5 Sistem Koordinat 3 Dimensi Kartensius	9
2.6 Regresi Linear	9
2.7 Uji Lack Of Fit	10
2.8 Perhitungan Kecepatan dan Percepatan	11
BAB III METODOLOGI PENELITIAN	13
3.1 Flowchart Metodologi Penelitian.....	13
3.2 Persiapan	15
3.3 Perancangan Skema Alat.....	15
3.4 Cara kerja	16
3.5 Proses Penggerjaan	16
3.6. Mekanisme Fletcher's Trolley Sebagai Alat Verifikasi	17
3.7 Flowchart Metode Pendekripsi dan Jarak Piksel	20
3.8 Flowchart Kalibrasi Jarak Piksel Menjadi Jarak Sebenarnya	21
BAB IV PERANCANGAN DAN PEMBUATAN INSTRUMEN	25
4.1. Implementasi Program dalam Sistem	25

4.2. Konstruksi Program <i>Image Processing</i>	25
4.2.1. Program Pendekripsi Objek.....	26
4.2.2 Program Thresholding	28
4.2.3. Program Erode dan Dilate	30
4.2.4. Program Pendekripsi Objek.....	33
4.2.5. Program Perhitungan jarak piksel	35
4.2.6. Program Perhitungan jarak piksel menjadi jarak sebenarnya.....	37
4.2.7. Program Perhitungan kecepatan dan percepatan	39
4.3. Konsep Instrumen Penelitian	40
BAB V PENGUJIAN INSTRUMEN	43
5.1 Pendekripsi Objek.....	43
5.2 Pengukuran Jarak	43
5.3. Pengukuran Kecepatan dan Percepatan	50
5.3.1 Pengukuran Kecepatan	50
5.3.2. Pengukuran Percepatan.....	55
BAB VI PENUTUP	61
6.1. Kesimpulan	61
6.2. Saran dan Evaluasi.....	61
DAFTAR PUSTAKA	63
LAMPIRAN A	65
LAMPIRAN B	69
LAMPIRAN C	80
LAMPIRAN D	83
LAMPIRAN E	86
BIODATA PENULIS	89

DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Ilustrasi dari Pembentukan bayangan pada lensa kamera [5]	4
Gambar 2. 2 Warna RGB	6
Gambar 2. 3 Binary Image	7
Gambar 2. 4 Model warna HSV	8
Gambar 2. 5 Sistem Koordinat 3 Dimensi [8]	9
Gambar 2. 6 Pesebaran data dengan metode linier [9]	10
Gambar 3.1 Diagram alir metodologi penelitian.....	14
Gambar 3.2 Skema Alat Penelitian Pendekatan Perpindahan Posisi.....	16
Gambar 3.3. Ilustrasi dari fletcher's Trolley [13].....	17
Gambar 3.4. gaya – gaya pada fletcher's Trolley [13].....	18
Gambar 3.5. Fletcher's Trolley.....	19
Gambar 3.6. Ilustrasi Penelitian yang menggunakan fletcher's trolley.....	19
Gambar 3.7 Diagram alir metodologi pendekatan objek dan jarak piksel.....	20
Gambar 3.8 Diagram alir metodologi kalibrasi jarak piksel menjadi jarak sebenarnya.....	22
Gambar 4.1. Gambar RGB (1) sebagai citra masukan dirubah menjadi HSV (2).....	26
Gambar 4.2. Diagram alir proses pengolahan dari gambar RGB ke HSV.....	27
Gambar 4.3 Gambar HSV (1) menjadi Threshold (2).....	29
Gambar 4.4 Diagram alir proses threshold.....	29
Gambar 4.5. Gambar hasil threshold (1) yang telah diproses dengan erode-dilate (2).....	32
Gambar 4.6 Diagram alir proses erode dan dilate///.....	32
Gambar 4.7 Trackbar yang digunakan untuk mencari perubahan gambar HSV menjadi gambar threshold.....	34
Gambar 4.8 Hasil nilai range HSV yang terdeteksi.....	35
Gambar 4.9 Diagram alir perhitungan jarak piksel.....	36

Gambar 4.10. Diagram Alir dari Perhitungan Jarak piksel menjadi jarak sebenarnya.....	38
Gambar 4.11 Diagram alir dari perhitungan kecepatan dan percepatan.....	40
Gambar 4.12 Desain sistem pendektsian posisi dari prototype turret gun.....	41
Gambar 4.13. Desain prototype turret gun.....	42
Gambar 5.1. Hasil Pendektsian Objek.....	43
Gambar 5.2. Ilustrasi perhitungan jarak piksel dalam gambar.....	44
Gambar 5.3 Grafik jarak piksel per 0.1 m.....	45
Gambar 5.4. Grafik perbandingan jarak asli dengan jarak yang terdeteksi oleh program.....	49
Gambar 5.5 Grafik Persentase error dari pengukuran jarak.....	50
Gambar 5.6. Grafik Perbandingan Garis Regresi tiap titik pendektsian kecepatan.....	54
Gambar 5.7. Grafik Perbandingan Garis Regresi tiap titik pendektsian percepatan.....	58

DAFTAR TABEL

Tabel 5. 1 Tabel Pengukuran jarak asli dengan jarak dalam program dan Error	48
Tabel 5. 2 Tabel Hasil Perhitungan Teori dari Kecepatan dan Percepatan	52
Tabel 5. 3 Tabel Perbandingan Kecepatan pada Program dan Kecepatan Teori	52
Tabel 5. 4 Tabel Hasil Error kecepatan setelah dikompensasi dengan metode regresi.....	55
Tabel 5. 5Tabel Perhitungan Teori dan Pendekripsi Percepatan	57
Tabel 5. 6 Tabel Hasil Error percepatan setelah dikompensasi dengan metode regresi.....	59

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengukuran merupakan metode dalam menentukan nilai besaran dari suatu objek. Proses pengukuran telah dilakukan seiring berkembangnya ilmu pengetahuan. Dalam pengukuran, *image processing* dapat digunakan untuk memperoleh data berupa jarak maupun posisi. Input yang berupa gambar atau video dapat diperoleh dengan menggunakan baik mono kamera maupun stereo kamera.

Image processing merupakan metode pengolahan data berupa gambar untuk mendapatkan gambar yang lebih disempurnakan atau untuk memperoleh informasi yang dibutuhkan. Dalam pengukuran, *image processing* dapat digunakan untuk memperoleh data berupa jarak maupun posisi. *Input* yang berupa gambar atau video dapat diperoleh dengan menggunakan baik mono kamera maupun stereo kamera. *Image processing* terbagi menjadi 3 tahap, yaitu *pre-processing*, *processing*, dan *post-processing*. *Pre-processing* adalah proses pengubahan warna HSV menjadi thresholding dimana gambar disegmentasi agar dapat membedakan objek yang akan diproses dengan latar belakang gambar. Kemudian *processing* merupakan tahap pendekripsi objek pada gambar. Tahap terakhir adalah *post-processing* dimana pengguna mengekstraksi informasi yang dibutuhkan dari gambar. Gambar yang diperoleh dari kamera diolah dengan bantuan perangkat lunak untuk memperoleh informasi yang dibutuhkan, misalnya jarak suatu objek dari kamera.

Tahun 2016 telah dikembangkan penggunaan teknologi pengolahan citra dengan memanfaatkan *stereo vision* dari dua kamera untuk menentukan koordinat 3 dimensi dari suatu objek. Metode pengolahan citra digunakan untuk mendekripsi objek dengan bantuan perangkat lunak, akan tetapi kamera ditempatkan pada posisi diam sehingga masih ditemukan *blindspot*, sehingga perlu adanya pengembangan agar tidak ditemukan *blindspot*. [1]

1.2 Rumusan Masalah

Dari uraian pada latar belakang dapat dirumuskan masalah sebagai berikut:

1. Bagaimana mengukur posisi dari suatu target dalam metode pengolahan citra?
2. Bagaimana cara mengukur kecepatan dari suatu objek dalam metode pengolahan citra?
3. Bagaimana cara mengukur percepatan dari suatu objek dalam metode pengolahan citra?

1.3 Batasan Masalah

Untuk menyederhanakan tugas akhir ini, maka batasan masalah yang akan dibahas antara lain:

1. Menggunakan kamera.
2. Jarak maksimal pendeksiyan sebesar 3 m.
3. Menggunakan *laser pointer* berwarna hijau.

1.4 Tujuan Penelitian

Adapun tujuan dari tugas akhir ini adalah sebagai berikut:

1. Mengetahui cara pengukuran posisi suatu objek dengan menggunakan metode pengolahan citra dan memanfaatkan laser sebagai alat pembantu.
2. Mengetahui cara pengukuran kecepatan suatu objek dengan menggunakan metode pengolahan citra dan memanfaatkan laser sebagai alat pembantu.
3. Mengetahui cara pengukuran percepatan suatu objek dengan menggunakan metode pengolahan citra dan memanfaatkan laser sebagai alat pembantu.

1.5 Manfaat Penelitian

Manfaat dari tugas akhir ini adalah sebagai berikut:

1. Dapat digunakan sebagai alat pengukuran
2. Dapat dijadikan sebagai referensi dalam bidang *image processing* dalam penelitian berikutnya
3. Dapat digunakan untuk pengembangan otomatisasi mesin-mesin militer.

BAB II

DASAR TEORI

2.1 Penelitian Sebelumnya

Penelitian sebelumnya telah dikembangkan penggunaan teknologi pengolahan citra dengan memanfaatkan *stereo vision* dari dua kamera untuk menentukan koordinat 3 dimensi dari suatu objek. Metode pengolahan citra digunakan untuk mendeteksi objek dengan bantuan perangkat lunak, akan tetapi kamera ditempatkan pada posisi diam sehingga masih ditemukan *blindspot*, sehingga perlu adanya pengembangan agar tidak ditemukan *blindspot*. Tingkat keberhasilan program sebesar 94,6%. Program pendekripsi koordinat z, x, dan y masing-masing memiliki nilai error terbesar 3,92%, 0,64%, dan 2,5%. [1]

Penelitian untuk pendekripsi jarak dengan metode *image processing* telah dikembangkan lebih lanjut dengan pendekripsi terhadap objek berbahaya secara *real time* dengan *mono camera*. Jarak dihasilkan dengan menggunakan metode pengukuran luas dari cahaya yang dihasilkan lampu LED dari objek. Hasil error pada percobaan diperoleh 5,51%. Sehingga user dapat mengetahui jarak sebuah objek dengan menggunakan sebuah kamera pada objek berbahaya. [2]

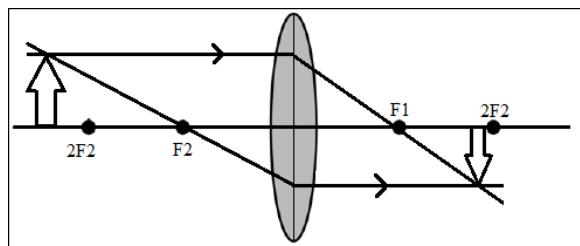
Penelitian untuk pendekripsi jarak objek dengan metode *image processing* telah dikembangkan dengan melakukan analisis geometri dengan menggunakan teorema pythagoras dari pengukuran jarak dengan menggunakan *mono-camera* dan laser. Laser yang diarahkan pada objek tertentu lalu terdeteksi oleh kamera lalu dengan persamaan geometri antara laser dengan kamera diperoleh nilai jaraknya dengan menggunakan bantuan perangkat lunak. Sinar laser yang diarahkan pada objek tertentu dengan kondisi jarak antara laser dan webcam 6 cm memiliki tingkat akurasi paling bagus pada jarak 22 cm, 40 cm dan 80 cm dengan rata-rata error sebesar 0,11%. [3]

Pengembangan metode *image processing* untuk pendekripsi kecepatan objek telah dikembangkan dengan metode perubahan frame gambar. *frame* yang terdeteksi dari gambar per detiknya

sehingga diperoleh informasi untuk menentukan ada apa tidaknya suatu gerakan sehingga, diperoleh besaran kecepatan. Informasi diperoleh dari tingkat kecepatan sebuah kendaraan yang lewat pada lintasan jalan. Kamera diletakkan pada tiang dengan ketinggian \pm 5,5 meter dan mengatur posisi sudut kamera dengan memiliki sudut sebesar 60° . Hasil perhitungan kecepatan kendaraan pada sistem mempunyai nilai standar deviasi sebesar 2,82 km/jam dan nilai ketidakpastian ± 1 km/jam. [4]

2.2 Kamera

Kamera merupakan alat optik yang dapat digunakan untuk menyimpan atau merekam bayangan dalam bentuk gambar foto. Kamera terdiri atas lensa cembung. Cahaya yang masuk pada kamera diatur oleh lebar celah diafragma atau aperture dan film negative untuk menangkap cahaya. Jika dihubungkan dengan bagian mata, ada keserasian antara mata dan kamera, yaitu diafragma seperti iris dan negatif film seperti retina mata. Karakteristik dari kamera antara lain, jarak fokus tetap, mengatur jarak benda maupun jarak bayangannya dengan menggeser-geser bayangan. Bayangan yang diperoleh bersifat nyata, terbalik, dan diperkecil. [5]



Gambar 2. 1 Ilustrasi dari Pembentukan bayangan pada lensa kamera [5]

Karena kamera menggunakan lensa cembung, maka rumus untuk mendapatkan fokus lensa cembung adalah sebagai berikut:

$$\frac{1}{f} = \frac{1}{s} + \frac{1}{s'} \quad (1)$$

dimana, f adalah fokus lensa cembung, s adalah jarak objek dan s' adalah jarak bayangan dari objek. Lalu untuk mendapatkan pembesaran dari lensa cembung pada kamera adalah sebagai berikut:

$$M = \frac{s}{s'} \quad (2)$$

dimana M , adalah pembesaran dari lensa cembung. [5]

2.3 Gambar Digital

Gambar digital merupakan citra $[m, n]$ yang dijelaskan dalam ruang *discrete* 2D yang berasal dari citra analog 2D (x, y) melalui proses pengambilan sampel yang disebut sebagai digitalisasi. Citra *continue* 2D $a(x, y)$ dibagi menjadi baris N dan kolom M . Itu persimpangan baris dan kolom disebut piksel.

Sebenarnya, dalam banyak kasus Citra analog (x, y) yang biasa dianggap sebagai fisik sinyal yang menimpa sensor 2D, sesungguhnya merupakan fungsi dari banyak variabel termasuk kedalaman (z) , warna (λ) , dan waktu (t) . Fungsi intensitas cahaya dapat ditulis sebagai $f(x, y)$ yang mana f merupakan nilai amplitude pada koordinat x dan y . Secara matematis, gambar digital dapat dituliskan sebagai matriks, seperti ditunjukkan pada persamaan [6]:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N - 1) \\ f(1,0) & f(1,1) & \dots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1,0) & f(M - 1,1) & \dots & f(M - 1, N - 1) \end{bmatrix} \quad (3)$$

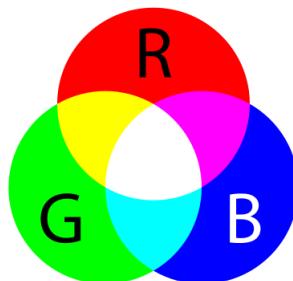
2.4 Kedalaman Warna

Proses yang dijalani otak dalam menerima dan menginterpretasi warna adalah sebuah fenomena alam yang masih belum bisa dijelaskan. Bentuk fisik dari warna bisa dilihat dari hasil teoritis dan eksperimen. Umumnya, warna yang mampu dideteksi oleh makhluk hidup berupa cahaya yang terpantul dari sebuah objek. Karakteristik dari cahaya merupakan ilmu dari pembeda warna, apabila cahaya tersebut akromatis, maka yang ditampilkan

adalah hitam dan putih dan pasca-indera dari sebuah mata mampu membedakan warna umum yakni merah, hijau dan biru. [7]

1. Gambar Berwarna (RGB)

Gambar berwarna merupakan gambar yang memiliki informasi warna di setiap pikselnya. Informasi warna ini dapat dibentuk dari gabungan komponen-komponen satu set *channel* warna. Gambar berwarna memiliki format gambar RGB (*Red Green Blue*) yang mana warna yang dihasilkan tiap pikselnya terdiri dari gabungan informasi tingkatan nilai dasar merah, hijau dan biru. [7]



Gambar 2. 2 Warna RGB

RGB adalah model warna dimana gambar terdiri dari tiga citra independen yaitu merah, hijau, dan biru seperti yang ditunjukkan pada Gambar 2.2. Dalam menentukan warna tertentu, masing-masing nilai menunjukkan jumlah masing-masing komponen yang ada pada setiap piksel, dengan nilai yang lebih tinggi sesuai dengan piksel yang lebih terang. Ruang warna ini banyak digunakan karena sesuai dengan tiga fotoreseptor mata manusia. [7]

2. Gambar Biner

Gambar biner merupakan gambar yang hanya memiliki dua macam warna, yaitu putih dan hitam. Pada gambar biner, setiap piksel hanya memiliki 2 macam nilai, yaitu 1 untuk warna hitam dan 0 untuk warna putih. Pada gambar 2.3 dapat dilihat gambar dengan format biner, yang mana gambar hanya memiliki warna

hitam dan putih. Gambar biner pada umumnya diperoleh melalui proses thresholding. [7]

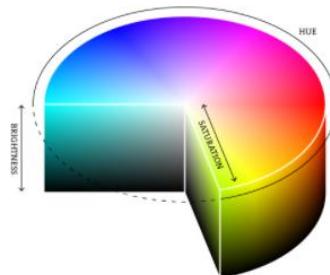


Gambar 2. 3 Binary Image

Thresholding merupakan proses segmentasi gambar. Segmentasi gambar sendiri digunakan untuk membedakan antara objek dengan latar belakangnya. Proses segmentasi sendiri dilakukan dengan mengisolasi objek pada gambar yang berbentuk format *grayscale* menjadi format biner. Pada saat melakukan proses *thresholding* perlu diperhatikan nilai *threshold* yang akan ditentukan. Nilai piksel yang berada di bawah nilai *threshold* akan diubah menjadi warna hitam, sedangkan piksel yang memiliki nilai di atas nilai *threshold* akan diubah menjadi warna putih. [7]

3. Gambar HSV

Model warna lain yang dapat dimanfaatkan adalah *hue*, *saturation*, *value* (HSV). HSV banyak digunakan untuk membuat perbandingan warna karena Hue adalah variasi cahaya yang hampir independen. [7]



Gambar 2. 4 Model warna HSV

Gambar 2.4 berikut menunjukkan model warna ini mewakili masing-masing nilai warna sebagai bagian dari silinder, yang terdiri dari 3 komponen yaitu Hue, Saturation, Value. Dalam mengubah citra RGB menjadi citra HSV bisa digunakan persamaan seperti berikut ini:

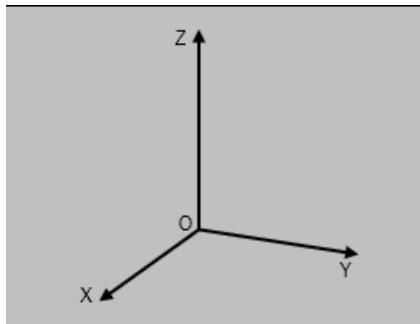
$$V = \max(r, g, b)$$

$$S = \begin{cases} 0, & v = 0 \\ 1 - \frac{\min(r, g, b)}{v}, & v > 0 \end{cases} \quad (4)$$

$$H = \begin{cases} 0, & \text{jika } S = 0 \\ \frac{60 * (g - b)}{s * v}, & \text{jika } v = r \\ 60 * \left[2 + \frac{b - r}{s * v} \right], & \text{jika } v = g \\ 60 * \left[4 + \frac{r - g}{s * v} \right], & \text{jika } v = b \end{cases} \quad (5)$$

Jika Saturation $S=0$, maka hue tidak terdefinisi atau dengan kata lain tidak memiliki hue berarti *monochrome*. Hue (H) lalu dikonversi menjadi derajat/degrees dengan cara mengalikan dengan 60 sehingga menghasilkan HSV dengan S dan V antara 0 dan 1 dan H antara 0 – 360. [7]

2.5 Sistem Koordinat 3 Dimensi Kartensius

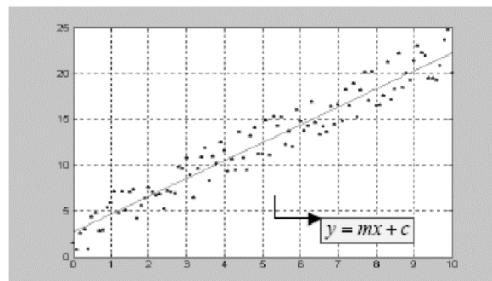


Gambar 2. 5 Sistem Koordinat 3 Dimensi [8]

Sistem koordinat kartesius merupakan sistem yang digunakan untuk menentukan posisi suatu titik, garis maupun gambar dalam suatu bidang baik 2 dimensi maupun 3 dimensi. Dalam sistem koordinat 2 dimensi, digunakan dua garis yang saling tegak lurus dalam satu bidang sebagai sumbu koordinat. Sumbu yang ke arah horizontal diberi label axis-x, sedangkan sumbu yang ke arah vertikal diberi label axis-y. Dalam bidang 3 dimensi, ditambahkan satu sumbu lagi yang arahnya tegak lurus terhadap bidang xy. Sumbu yang tegak lurus terhadap bidang xy ini diberi label sumbu z. Sumbu-sumbu pada koordinat kartesian ini bersifat orthogonal (antar sumbu saling tegak lurus). [8]

2.6 Regresi Linear

Regresi adalah sebuah teknik untuk memperoleh persamaan kurva pendekatan dari titik-titik data. Regresi linier digunakan menentukan fungsi linier (garis lurus) yang paling sesuai dengan kumpulan titik data (x_n, y_n) yang diketahui. [9]



Gambar 2. 6 Pesebaran data dengan metode linier [9]

Dalam regresi linier ini yang dicari adalah nilai m dan c dari fungsi linier $y=mx+c$, dengan:

Adapun langkah langkah untuk menentukan dari algoritma regresi linier adalah sebagai berikut:

$$m = \frac{N \sum_{n=1}^N x_n y_n - \left(\sum_{n=1}^N x_n \right) \left(\sum_{n=1}^N y_n \right)}{N \sum_{n=1}^N x_n^2 - \left(\sum_{n=1}^N x_n \right)^2}$$

$$c = \frac{\sum_{n=1}^N y_n}{N} - m \frac{\sum_{n=1}^N x_n}{N} = \bar{y} - m\bar{x} \quad (7)$$

1. Tentukan N titik data yang diketahui dalam untuk $i=1,2,3,\dots,N$ (x_i,y_i)
2. Hitung nilai m dan c dengan menggunakan formulasi dari regresi linier di atas
3. Tampilkan fungsi linier
4. Hitung fungsi linier tersebut dalam range x dan step dx tertentu
5. Tampilkan hasil tabel (x_n,y_n) dari hasil fungsi linier tersebut.[8]

2.7 Uji Lack Of Fit

Lack of fit artinya penyimpangan atau ketidak tepatan terhadap model linier order pertama. Pengujian lack of fit artinya pengujian untuk mendeteksi apakah model linier order pertama tepat. Bila lack of fit tidak bermakna maka model linier order

pertama tepat, sedangkan apabila lack of fit bermakna maka model linier order pertama tidak tepat, perlu dikembangkan menjadi model linier kuadratik atau model nonlinier. Pengujian lack of fit ini diperlukan bila terdapat *pengamatan berulang*, yaitu satu nilai prediktor a-tau satu kombinasi nilai prediktor (bila digunakan beberapa prediktor) yang berpasangan dengan beberapa nilai respon. [10]

Untuk mengetahui keakuratan dari model yang diperoleh dengan data, maka dibandingkan dengan p-value dari model tersebut ke tingkat signifikan yang telah ditentukan. Tingkat signifikan (α) adalah sebuah peluang dari data untuk menyimpulkan model tidak sesuai dengan data, Hipotesis :

- H_0 : model linear orde pertama tepat
- H_1 : model linear orde pertama tidak tepat
- $P\text{-value} < \alpha$, H_0 ditolak
- $P\text{-value} > \alpha$, H_0 gagal ditolak.

2.8 Perhitungan Kecepatan dan Percepatan

1. Perhitungan Kecepatan

Kecepatan rata-rata ditentukan berdasarkan perpindahan benda. Perpindahan benda adalah vektor pengubung posisi awal dengan posisi akhir sehingga selalu berupa garis lurus. Perpindahan benda tidak menyatakan lintasan yang ditempuh benda. Panjang lintasan yang ditempuh benda lebih besar atau sama dengan panjang perpindahan benda. Panjang lintasan dan besar perpindahan sama hanya jika benda bergerak dalam garis lurus. [11]

Kecepatan rata-rata adalah rasio antara jarak tempuh dengan waktu tempuh. Karena jarak tempuh umumnya lebih besar daripada besar perpindahan maka laju rata-rata umumnya lebih besar daripada besar kecepatan rata-rata. Laju rata-rata didefinisikan sebagai:

$$\nu = \frac{s}{\Delta t} \quad (8)$$

dimana, s adalah jarak tempuh dan Δt waktu tempuh. [11]

2. Perhitungan Percepatan

Selama gerakan kadang kecepatan benda berubah. Perubahan tersebut bisa berupa perubahan nilai saja, perubahan arah saja, atau perubahan nilai dan arah. Perubahan tersebut ada yang cepat dan ada yang lambat. Besaran yang mengukur berapa cepat kecepatan berubah dinamakan percepatan. Percepatan didefinisikan sebagai perbandingan antara perubahan kecepatan benda dengan lama kecepatan tersebut berubah. Misalkan saat t_1 kecepatan sesaat benda adalah v_1 dan pada saat t_2 kecepatan sesaat benda adalah v_2 , Maka

Perubahan kecepatan benda adalah

$$\Delta v_{21} = v_2 - v_1 \quad (9)$$

Lama waktu kecepatan berubah adalah

$$\Delta t_{21} = t_2 - t_1 \quad (10)$$

Definisi percepatan adalah

$$a = \frac{\Delta v_{21}}{\Delta t} \quad (11)$$

Percepatan juga merupakan besaran vector dan pada perhitungan percepatan kita tidak mempedulikan nilai kecepatan pada berbagai waktu. Yang dibutuhkan adalah kecepatan pada saat awal dan saat akhir.[11]

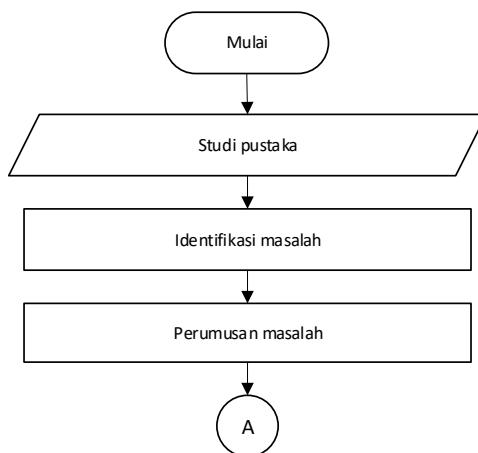
BAB III

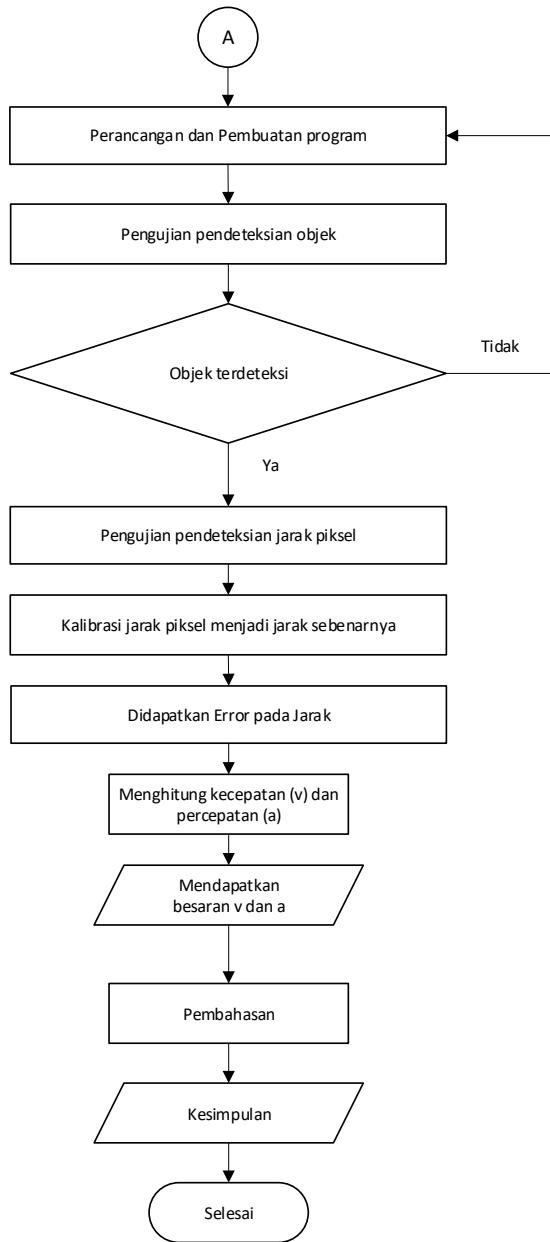
METODOLOGI PENELITIAN

Bab ini dijelaskan metode penelitian langkah-langkah sistematis sebagai acuan kerangka penelitian untuk pengujian pengukuran perubahan posisi dengan menggunakan metode *image processing* yang mana memanfaatkan *mono-camera* dan laser sebagai alat utama.

3.1 Flowchart Metodologi Penelitian

Metodologi penelitian yang digunakan untuk penelitian pengukuran posisi ini secara garis besar terdiri dari tiga proses utama, yaitu tahapan persiapan, tahap perancangan dan pembuatan program serta proses analisa data. Adapun proses persiapan terdiri dari studi pustaka, untuk mempelajari dasar-dasar penelitian ini, lalu proses identifikasi masalah serta perumusan masalah. Kemudian dilakukan proses perancangan program yang mana akan diperoleh hasil pendekripsi jarak piksel lalu dikalibrasikan menjadi jarak sebenarnya, lalu dikonversi mendapatkan nilai kecepatan dan percepatan.





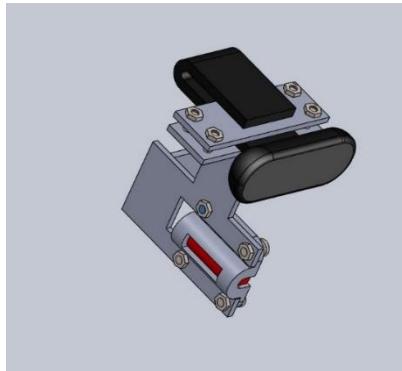
Gambar 3.1 Diagram alir metodologi penelitian

3.2 Persiapan

Proses persiapan terdiri dari tiga tahapan yaitu studi pustaka, identifikasi masalah dan perumusan masalah. Studi pustaka merupakan tahapan dimana acuan referensi yang penulis analisa akan digunakan sebagai pertimbangan dalam menyelesaikan permasalahan yang akan diteliti, yang mana dalam hal ini mengenai pembuatan aplikasi program pengukuran sumbu z dengan metode image processing. Studi pustaka dilakukan dengan memanfaatkan buku-buku referensi, paper-paper dan jurnal-jurnal yang berkaitan dengan permasalahan yang akan dibahas. Setelah itu dilakukan identifikasi masalah untuk menyusun program.

3.3 Perancangan Skema Alat

Pada proses ini dilakukan perancangan program kalibrasi mono kamera serta perancangan program untuk pendekripsi dan pengukuran objek terhadap satu kamera. Program yang akan dirancang penulis ini nantinya akan diaplikasikan pada prototype alat turret gun. Prototype turret gun terdiri menjadi beberapa bagian, seperti landasan utama, landasan rotasi, potensiometer, motor servo, laser, gir, kamera untuk mendekripsi jarak dan kamera untuk mendekripsi objek laser dan juga, kamera yang digunakan untuk pendekripsi optical flow. Prototype ini memiliki landasan utama yang terbuat dari akrilik setebal 3 mm yang berfungsi sebagai menjadi landasan dari alat penyangga dari landasan rotasi, di bawahnya landasan rotasi tersebut dipasangkan sebuah motor servo untuk memutar landasan rotasi searah sumbu x, lalu di landasan rotasi ada kamera yang bertujuan untuk mendekripsi optical flow yang di sumbu x dan y yang mana tidak akan dibahas di tugas akhir ini, lalu landasan rotasi ini dipasang sebuah dudukan gird an dudukan potensiometer dan motor servo, lalu pada gir tersebut dipasangkan dudukan laser dan dudukan kamera untuk mendekripsi laser yang mana bertujuan untuk menjadi pengukuran posisi hingga percepatan, yang mana menjadi input dari tugas akhir ini. Pada gambar kedua terdapat dudukan yang dirancang sendiri oleh peneliti. Kamera diberi jarak secara horizontal dengan laser sebesar 7 cm dan kamera diberi pengunci agar tidak bergerak



Gambar 3.2 Skema Alat Penelitian Pendekatan Perpindahan Posisi

3.4 Cara kerja

Cara kerja program yang harus dibuat adalah pertama mendekteksi objek yang terdeteksi laser berwarna merah dalam bentuk gambar 2D. kemudian data gambar yang terdeteksi kamera ini dianalisa menggunakan program C++ dari dengan bantuan software. Setelah itu dilakukan proses pendekatan jarak piksel, lalu dilakukan proses kalibrasi menjadi jarak sebenarnya dengan metode regresi lalu, setelah diperoleh jarak dilakukan perhitungan untuk mendapatkan kecepatan dan percepatan. Hasil jarak, kecepatan dan percepatan agar bisa diaplikasikan di prototype turrent gun.

3.5 Proses Penggeraan

Ada beberapa tahapan dalam proses dan penggeraan perancangan program yaitu pendekatan objek dan kalibrasi jarak asli dan sebenarnya dan perhitungan kecepatan dan percepatan.

1. Pendekatan Objek dan mendapatkan jarak piksel.

Pendekatan objek dilakukan dengan cara mengatur satu nilai warna (merah) dengan mengatur HSV yang ada. Setelah warna merah terdeteksi, lalu dilakukan pendekatan objek yang tertembak oleh warna laser tersebut. Data yang diperoleh adalah merupakan posisi objek dengan koordinat yang terdeteksi, lalu

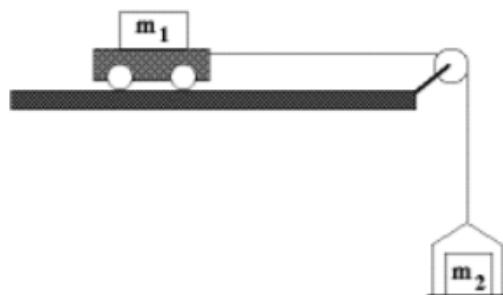
mencari jarak koordinat x dan y setelah itu digunakan perhitungan Pythagoras untuk mendapatkan jarak piksel

2. Perhitungan untuk jarak, kecepatan dan percepatan

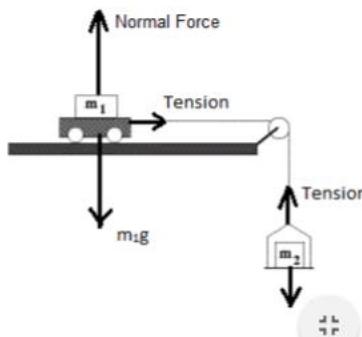
Perhitungan jarak digunakan dengan metode regresi linear setelah itu untuk mendapatkan kecepatan dilakukan dengan cara pendektsian perpindahan posisi yang dilakukan secara manual, lalu setelah itu dibagi dengan fungsi waktu pada program untuk mendapatkan kecepatan, setelah mendapatkan kecepatan lalu menggunakan fungsi waktu terhadap kecepatan untuk mendapatkan percepatan

3.6. Mekanisme Fletcher's Trolley Sebagai Alat Verifikasi

Penggunaan *fletcher's trolley* dalam praktikum fisika untuk mendapatkan nilai percepatan, sehingga pada penelitian ini mekanisme *fletcher's trolley* sebagai verifikasi dari hasil pendektsian dari kamera. Praktikum ini menggunakan 2 objek benda. Objek benda pertama ditempatkan di permukaan rata dan yang seimbang, objek kedua dipasangkan dengan tali yang juga mengikat objek pertama yang juga dihubungkan dengan katrol lalu, objek kedua dibiarkan menggantung, sehingga gravitasi menyebabkan objek pertama tertarik oleh tali dan objek kedua terjatuh, sehingga terjadi perpindahan posisi pada objek pertama, sehingga bisa diperhitungkan kecepatan dan percepatan yang dialami oleh objek pertama.[12]



Gambar 3.3. Ilustrasi dari *fletcher's Trolley* [12]



Gambar 3.4. gaya – gaya pada *fletcher's Trolley* [12]

Sehingga terlihat pada gambar 3.3 dan 3.4 bahwa gaya tarik pada tali bertindak secara horizontal, sehingga gaya normal dengan berat dari objek pertama akan berlawanan arah sehingga bernilai sama dan berselisih nol, lalu kedua gaya Tarik tali juga bersifat internal sehingga nilainya juga dianggap nol. Oleh karena itu, gaya total pada sistem *fletcher trolley* diperoleh dari objek kedua yang tergantung sehingga, diperoleh persamaan:

$$m_2g = (m_1 + m_2) \alpha \quad (12)$$

Dimana:

m_2 = massa objek kedua

m_1 = massa objek pertama

g = gravitasi ($9.81 \frac{m}{s^2}$)

α = nilai percepatan

Sehingga diperoleh nilai percepatan dari objek pertama, lalu untuk mendapatkan kecepatan dari objek adalah dengan rumus berikut:

$$v_x^2 = v_0^2 + 2\alpha s \quad (13)$$

Dimana:

s = nilai perpindahan

$$v_0 = \text{kecepatan awal}$$

$$v_x = \text{kecepatan akhir}$$

Hasil percepatan dan kecepatan yang dihasilkan dari *fletcher trolley* akan digunakan sebagai verifikasi dari perhitungan yang diperoleh dari program sehingga, akan ditemukan error, dalam memperhitungkan kecepatan dan percepatan yang terhitung dengan *fletcher's trolley* dengan kecepatan dan percepatan yang terdeteksi oleh program.

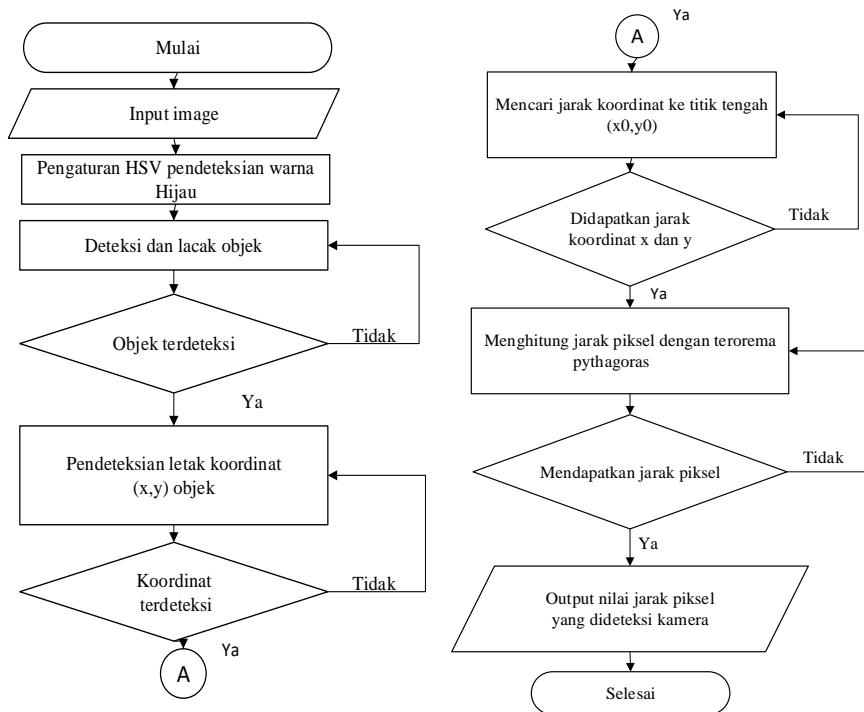


Gambar 3.5. Fletcher's Trolley



Gambar 3.6. Ilustrasi Penelitian yang menggunakan fletcher's trolley.

3.7 Flowchart Metode Pendeksiian dan Jarak Piksel

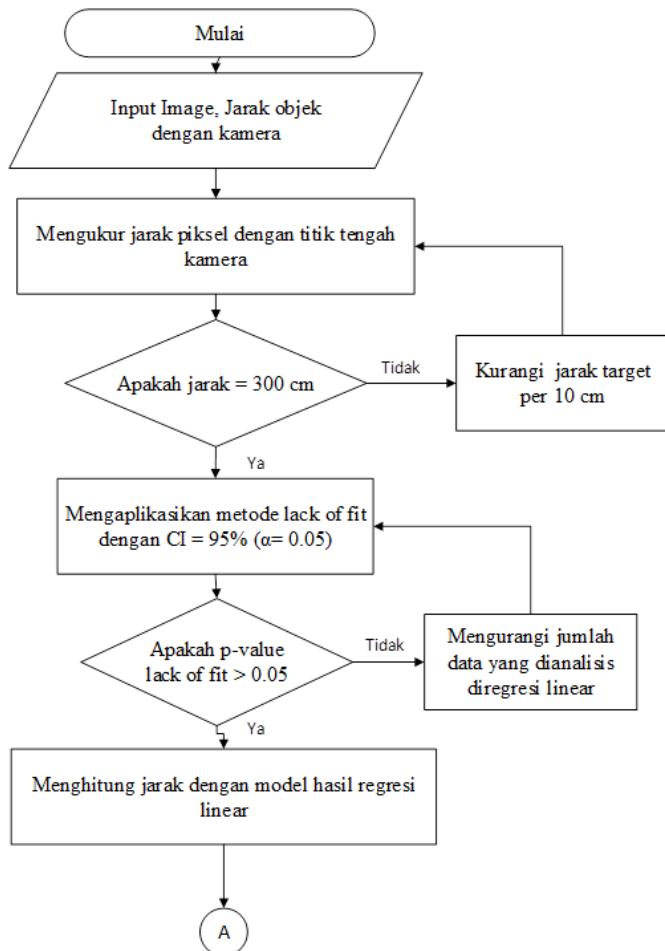


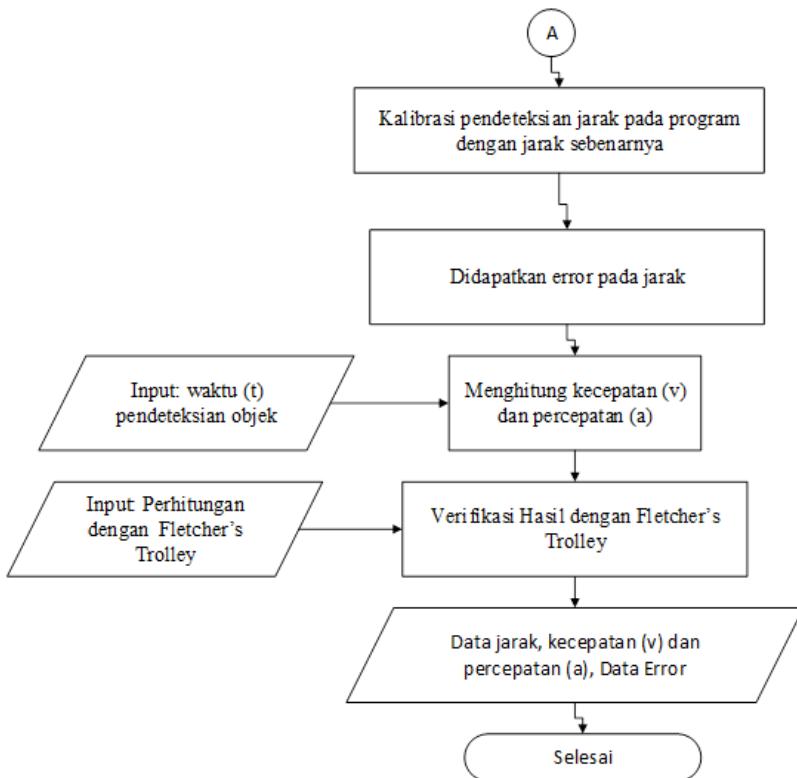
Gambar 3.7 Diagram alir metodologi pendeksiian objek dan jarak piksel

Diagram alir pada gambar 3.7 telah menunjukkan bahwa proses yang pertama kali dilakukan adalah dengan mendeksi objek ditembak oleh laser dengan menggunakan kamera. Gambar yang terdeteksi kamera lalu diatur tingkat HSVnya. Kemudian dari gambar tersebut dilakukan pendeksiian dari warna merah laser yang ditembakkan ke arah objek, lalu setelah diperoleh posisi warna merah pada objek tersebut lalu dicari koordinat x dan y. setelah mendapatkan koordinat x dan y maka dilakukan pencarian jarak koordinat x dan y dengan mengurangi koordinat x dan y yang terdeteksi dengan titik tengah dari kamera (x_0, y_0). Lalu setelah

diperoleh jarak koordinat digunakan terorema Pythagoras untuk mendapatkan jarak sebenarnya.

3.8 Flowchart Kalibrasi Jarak Piksel Menjadi Jarak Sebenarnya





Gambar 3.8 Diagram alir metodologi kalibrasi jarak piksel menjadi jarak sebenarnya

Dari diagram alir pada gambar 3.8 bahwa proses yang pertama kali dilakukan adalah menginput objek dan mengukur jarak piksel dengan titik tengah kamera., lalu ketika jarak sudah terdeteksi hingga jarak maksimal (3 m), maka diperoleh data berupa jarak piksel per 0.1 m. Setelah diperoleh jarak piksel, data dianalisis dengan metode regresi linear dan menggunakan metode *lack of fit* untuk mendekripsi keakuratan model linear. Adapun *Confidence Interval* yang digunakan adalah sebesar 95% dengan $\alpha = 0.05$. Setelah dilakukan metode *lack of fit*, maka akan menghasilkan p-value, apabila p-value < 0.05 maka data tersebut ditolak dikarenakan model regresi yang dihasilkan tidak sesuai,

maka harus dilakukan pengujian data ulang dengan mengurangi jumlah data yang sekiranya tidak linear dan apabila, p-value > 0.05, maka data tersebut gagal tolak, dan bisa melakukan perhitungan dengan model regresi untuk mendapatkan jarak yang dideteksi oleh program. Langkah selanjutnya adalah kalibrasi dengan jarak sebenarnya, lalu nanti diperoleh error. Lalu setelah mendapatkan error dilanjutkan dengan perhitungan kecepatan dan percepatan, Lalu dilanjutkan dengan verifikasi dengan menggunakan fletcher's trolley . Setelah dilakukan verifikasi maka akan ditemukan error dalam pendekstrian kecepatan dan percepatan

(Halaman ini sengaja dikosongkan)

BAB IV

PERANCANGAN DAN PEMBUATAN INSTRUMEN

4.1. Implementasi Program dalam Sistem

Program pendektsian perpindahan posisi dibuat dengan menggunakan teknologi computer vision. Perangkat lunak yang digunakan dalam pembuatan program adalah Microsoft Visual Studio Express 2013 dan menggunakan database OpenCV 2.4.10. Mesin pengolah yang digunakan untuk menjalankan program adalah personal computer dengan spesifikasi sebagai berikut:

- a. Dell Inspiron Pandora 7447
- b. Processor Intel® Core™ i7-4720HQ CPU @ 2.6GHz
- c. Memory 8192 MB RAM
- d. Operating System Windows 10 Home (64-bit)

Untuk mengetahui performa dari program pendektsian perpindahan posisi dengan bantuan laser. Maka perlu dilakukan pengujian. Pengujian dilakukan pada kondisi tertentu. Laser yang digunakan adalah laser berwarna merah yang ditembakkan ke objek tertentu. Pengambilan data dilakukan didalam ruangan dengan pencahayaan konstan.

4.2. Konstruksi Program *Image Processing*

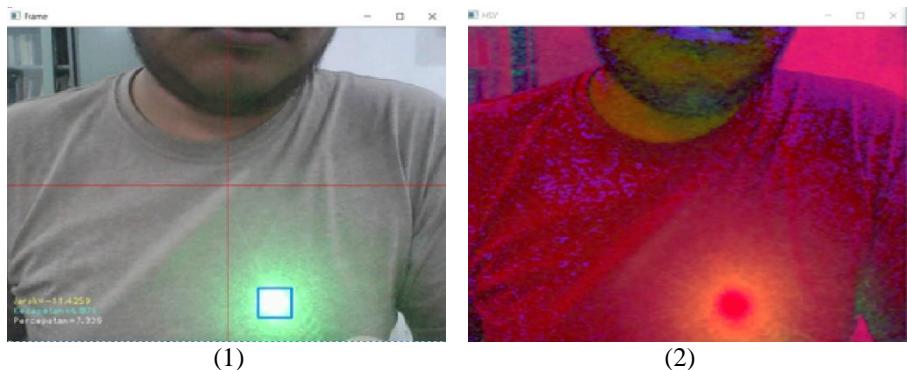
Dalam mengkonstruksi program dibutuhkan perangkat lunak sebagai landasan pembuatan program yang disesuaikan dengan metode pengukuran perpindahan dengan menggunakan laser. Adapun perangkat lunak yang digunakan pada pembuatan program ini adalah Visual Studio Express 2013 dengan menggunakan bahasa C++, dengan menggunakan library OpenCV 2.4.10 sebagai pustaka pengambilan fungsi perintah pengolahan citra. Dalam merancang suatu program perlu diperhatikan bahwa banyak nya proses yang dilakukan oleh program akan mempengaruhi lamanya proses penggeraan, seperti dalam menggunakan OpenCV sebagai pre-processor sebelum akhirnya diproses kedalam editor code yang akan di-compile menjadi executable program.

4.2.1. Program Pendeksiian Objek

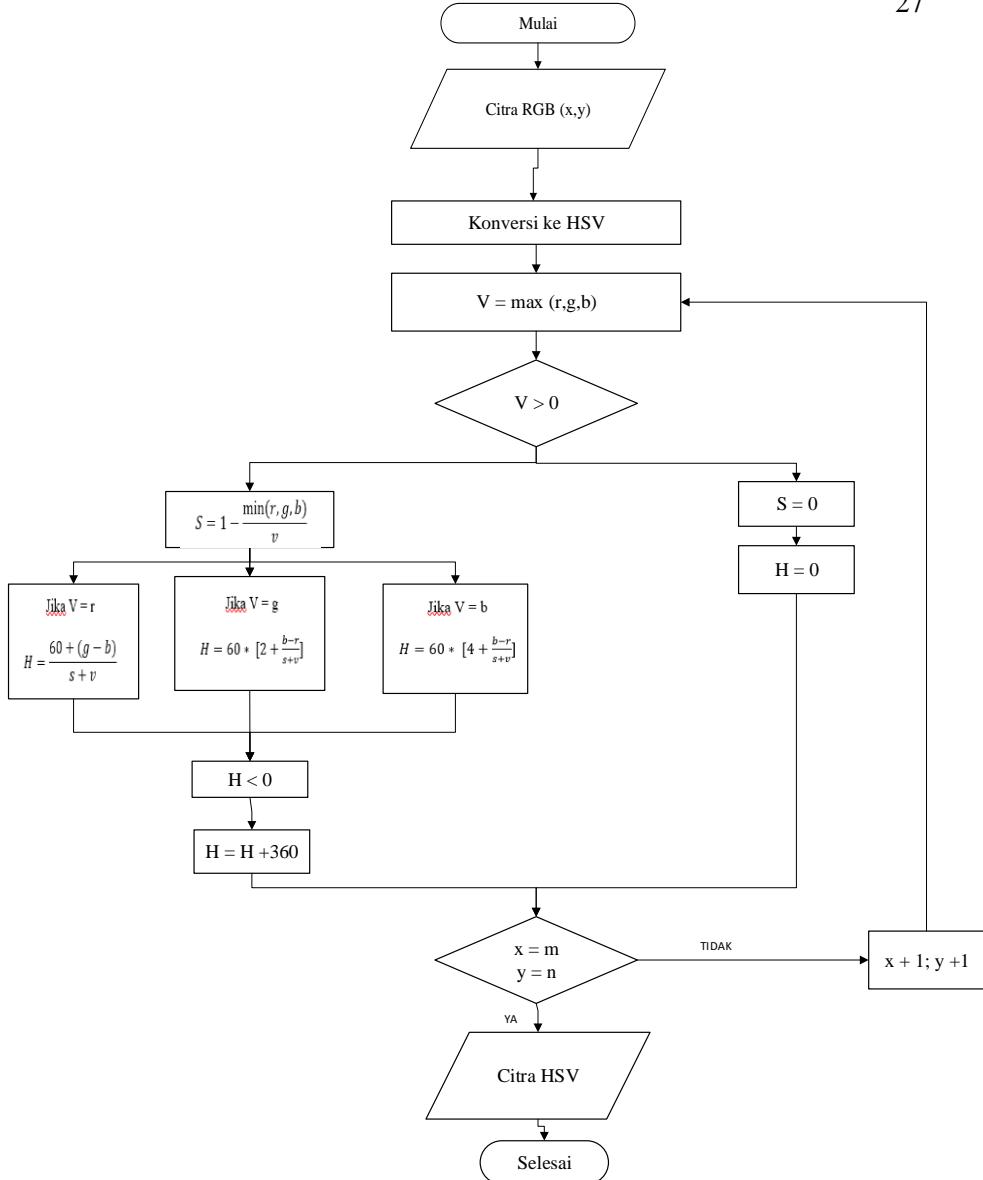
Kode Program 1: RGB to HSV

```
Mat img, img_gray, channel[3], img_HSV;  
cvtColor(img, img_HSV, COLOR_BGR2HSV);
```

Tahap pertama yang dilakukan adalah menginput gambar secara real-time dari kamera kedalam memori pengolahan data dalam bentuk matriks pixel sebesar 640 lebar x 480 tinggi pixel, saat proses ini gambar yang ditangkap oleh kamera secara otomatis merupakan gambar format RGB. Setelah gambar berhasil di ambil, selanjutnya gambar di konversi kedalam format HSV. Hal ini dikarenakan pendeksiian gambar yang dilakukan dalam format HSV lebih baik dalam menentukan objek jika dibandingkan dengan pendeksiian pada format RGB.



Gambar 4.1. Gambar RGB (1) sebagai citra masukan dirubah menjadi HSV (2)



Gambar 4.2. Diagram alir proses pengolahan dari gambar RGB ke HSV

Frame hasil dari tangkapan kamera Logitech C-170 berbentuk format gambar RGB (Red Green Blue) dengan resolusi 640x480 pixel seperti yang terlihat pada Gambar 4.1. Hasil dari frame tersebut akan diproses menjadi citra HSV. Untuk merubah dari gambar RGB ke HSV secara manual dapat digunakan persamaan (3). Gambar 4.2 adalah diagram alir dari proses pengolahan citra RGB ke HSV, sedangkan pada Gambar 4.1 dapat dilihat proses perubahan dari citra RGB menjadi citra HSV.

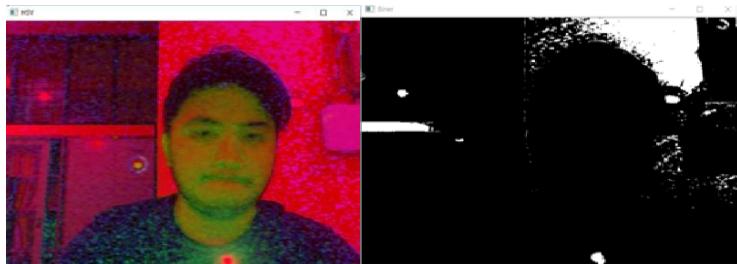
4.2.2 Program Thresholding

Kode Program 2: *Threshold*

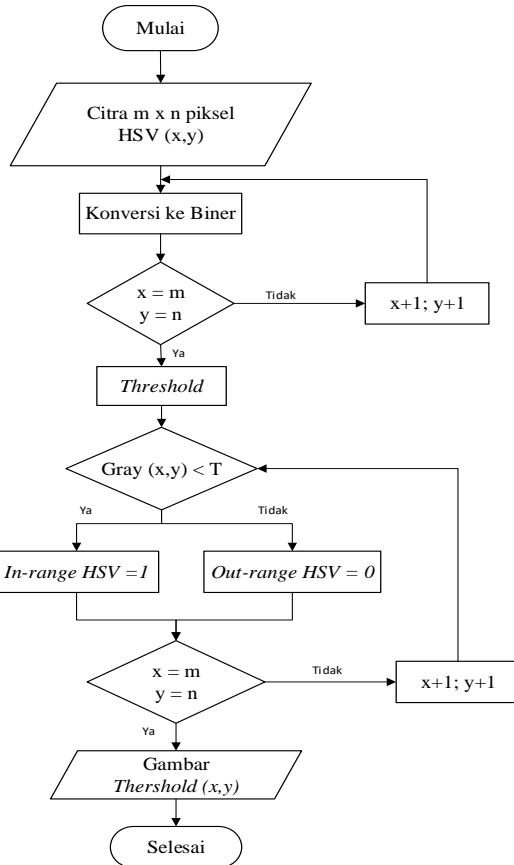
```
//Threshold the image
Mat imgThresholded;
Mat imgThresholded2;
inRange(img_HSV, Scalar(iLowH, iLowS, iLowV),
Scalar(iHighH, iHighS, iHighV), imgThresholded);
inRange(img_HSV, Scalar(iLowH, iLowS, iLowV),
Scalar(iHighH, iHighS, iHighV), imgThresholded2);
img_gray = imgThresholded;
```

Proses kedua dari tahap pendektsian objek adalah thresholding pada gambar HSV. Tipe threshold yang digunakan adalah binary threshold dengan range nilai HSV minimum (0, 0, 0) - maksimum (255, 255, 255) baik untuk HSV low dan HSV high. Pada gambar 4.4 dapat dilihat diagram alir dari proses threshold pada gambar HSV.

Hasil perubahan gambar HSV menjadi gambar biner dengan tipe binary threshold dapat dilihat pada Gambar 4.3. Pada gambar 4.3 dapat dilihat bahwa warna putih menunjukan objek yang ingin di deteksi dan warna hitam merupakan background dari objek. Namun terkadang hasil yang diperoleh kurang sempurna. Pada gambar 4.3 dapat dilihat bahwa bentuk hasil threshold dari gambar LED masih kurang sempurna.



(1) (2)
Gambar 4.3 Gambar HSV (1) menjadi Threshold (2)



Gambar 4.4 Diagram alir proses *threshold*

4.2.3. Program Erode dan Dilate

Kode Program 3: *Erode and Dilate, edge detection dan pendeksiian*

```

erode(img_gray, img_gray, Mat(), Point(-1,-1), 4);
dilate(img_gray, img_gray, Mat(), Point(-1,-1), 4);
erode(img_gray, img_gray, Mat(), Point(-1,1), 4);
            dilate(img_gray, img_gray, Mat(), Point(-1,1),
4);

        vector<vector<Size>> contors;
        vector<Vec4i> heirarchy;
        findContours(img_gray, contors, heirarchy,
CV_RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE, Point(0,0));

        vector<Rect> boundRect(contors.size());
        vector<vector<Point>>
contor_poly(contors.size());
for(int i =0; i< contors.size(); i++)
{
    approxPolyDP(Mat(contors[i]),
contor_poly[i], 3, true);
    boundRect[i] =
boundingRect(Mat(contor_poly[i]));
}
int max_index = 0,max_area = 0;
for(int i =0; i< boundRect.size(); i++)
{
    int a = boundRect[i].area();
    //rectangle(img, boundRect[i].tl(),
boundRect[i].br(), Scalar(255,255,0), 2, 8, 0);
    if( a < max_area)
    {
        max_area = a;
        max_index = 1;
    }
}

```

```

    }

        int confidence = 0;
        for(int i=0; i< boundRect.size(); i++)
        {
            if((boundRect[i].x <
            boundRect[max_index].x + boundRect[max_index].width
            && boundRect[i].x > boundRect[max_index].x -
            int(0.1*boundRect[max_index].width)) &&
            (boundRect[i].y > boundRect[max_index].y) )
                confidence += 45;

        }
        if(boundRect.size() > 0)
        {
            if(confidence > 99)
                confidence = 0;
            try{
                Mat sub_image = Mat(img,
Rect(max(boundRect[max_index].x-30, 0),
max(boundRect[max_index].y-30, 0),
min(int(boundRect[max_index].width*1.75), img.cols -
boundRect[max_index].x+30),
min(boundRect[max_index].height*3, img.rows -
boundRect[max_index].y+30)));
                //imshow("Frame", sub_image);
            }catch(int e){
                cout<<"Error
occured"<<endl;
            }
            rectangle(img,
            boundRect[max_index].tl(), boundRect[max_index].br(),
Scalar(255,140,0), 2, 8, 0);
        }
    }
}

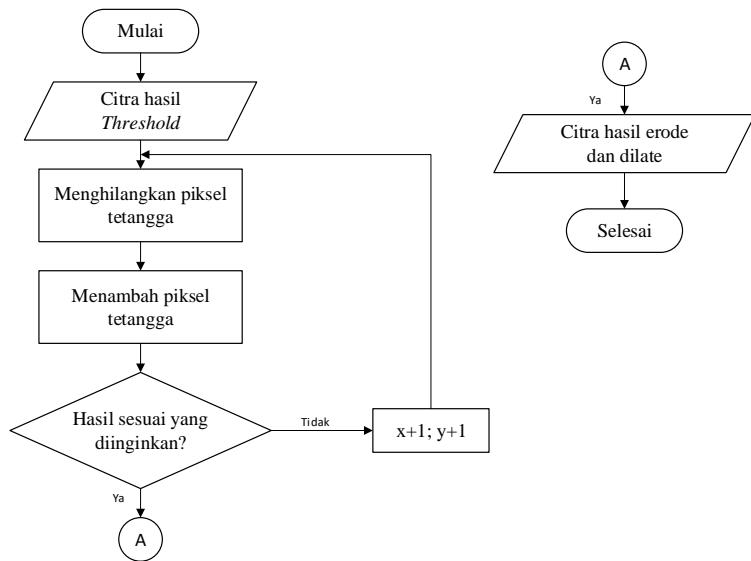
```

Pada proses threshold ini diperlukan juga proses noise filtering. Diagram alir dari proses noise filtering dapat dilihat pada Gambar 4.6. Proses ini dilakukan karena pada proses threshold biasanya terdapat gangguan/noise pada hasil gambar yang diperoleh. Noise filtering bertujuan untuk membuang atau

mengurangi gangguan tersebut agar gambar threshold yang diperoleh sesuai dengan yang diinginkan. Proses noise filtering terdiri dari erode dan dilate, dimana erode bertujuan untuk mengurangi pixel tetangga sedangkan dilate bertujuan untuk menambah pixel tetangga.



Gambar 4.5. Gambar hasil *threshold* (1) yang telah diproses dengan erode-dilate (2)



Gambar 4.6 Diagram alir proses erode dan dilate

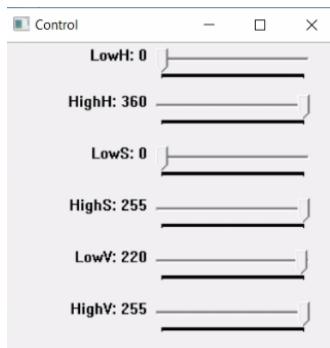
Pada Gambar 4.5 dapat dilihat perubahan pada gambar threshold yang dilakukan proses erode. Erode dilakukan dengan pengecilan pixel yang merusak. Citra threshold yang sudah di-erode kemudian dilakukan dilate dengan tiap 5 pixel tetangga dari pixel terluar. Hasil dari dilate tersebut dapat dilihat seperti pada Gambar 4.5.

4.2.4. Program Pendeteksian Objek

Kode Program 4 : Kalibrasi untuk Mencari Range Warna Objek

```
//create a window called "Control"
namedWindow("Control", CV_WINDOW_AUTOSIZE);
/*Create trackbars in "Control" window*/
cvCreateTrackbar("LowH", "Control", &iLowH, 360);
//Hue (0 - 179) 26-125
cvCreateTrackbar("HighH", "Control", &iHighH, 360);
cvCreateTrackbar("LowS", "Control", &iLowS, 255);
//Saturation (0 - 255) 83-255
cvCreateTrackbar("HighS", "Control", &iHighS, 255);
cvCreateTrackbar("LowV", "Control", &iLowV, 223); //Value(0 - 255) 85
cvCreateTrackbar("HighV", "Control", &iHighV, 255);
```

Proses berikutnya adalah mencari nilai range HSV untuk warna objek yang akan dideteksi pada gambar yang telah berformat HSV. Pada gambar 4.6 dapat dilihat trackbar yang digunakan untuk mencari nilai range HSV untuk mengubah gambar HSV menjadi gambar threshold.

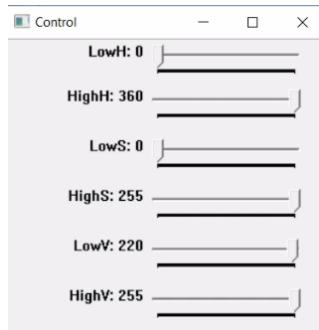


Gambar 4.7 Trackbar yang digunakan untuk mencari perubahan gambar HSV menjadi gambar threshold

Lalu pada, Gambar 4.7 dapat dilihat nilai range HSV yang diperoleh untuk mendeteksi objek. Nilai range HSV tersebut adalah sebagai berikut:

- H low : 0
- S low : 0
- V low : 220
- H high : 255
- S high : 255
- V high : 255

Nilai range HSV yang diperoleh akan digunakan untuk pendekstrian objek. Penentuan range HSV dilakukan beberapa kali percobaan untuk mendapatkan hasil pendekstrian objek yang diinginkan. Kondisi lingkungan seperti intensitas cahaya saat dilakukan proses pendekstrian juga sangat berpengaruh, hal ini menyebabkan nilai range HSV yang diperoleh dapat berubah ketika intensitas cahaya pada lingkungan tempat pengujian berubah.



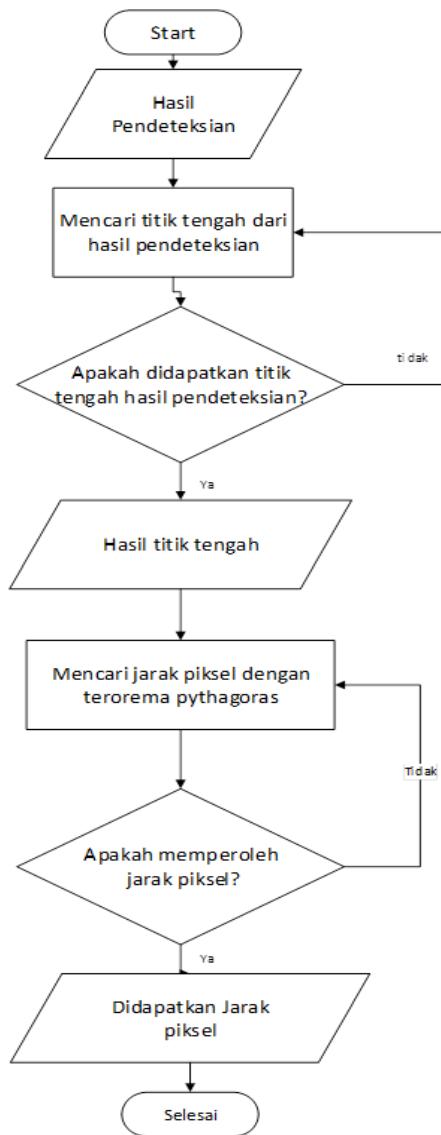
Gambar 4.8 Hasil nilai range HSV yang terdeteksi.

4.2.5. Program Perhitungan jarak piksel

Kode Program 5: Perhitungan jarak piksel

```
t_tengahx = boundRect[max_index].x +
(boundRect[max_index].width / 2);
t_tengahy =
boundRect[max_index].y +
(boundRect[max_index].height / 2);
jaraky = t_tengahy - 240;
jarakx = t_tengahx - 320;
pitago = sqrt((jarakx *
jarakx) + (jaraky * jaraky));
```

Proses selanjutnya setelah pendektsian piksel adalah menghitung jarak piksel, Jarak piksel dihitung dari titik tengah dari persegi yang terbentuk dari hasil pendektsian, lalu setelah memperoleh titik tengah dari hasil pendektsian dilakukan perhitungan dari titik tengah kamera dengan dua sumbu yakni, sumbu x dan sumbu y, setelah itu dilakukan perhitungan dengan terorema Pythagoras agar mendapatkan mendapatkan jarak piksel dari titik tengah pada gambar. Diagram perhitungan jarak piksel bisa dilihat pada gambar 4.9



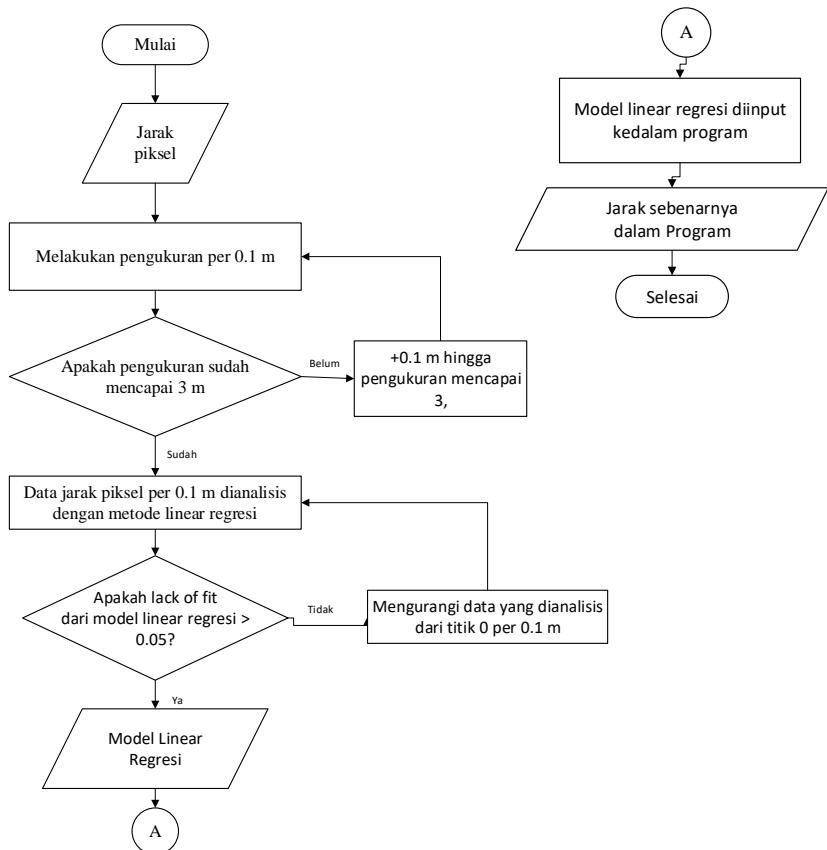
Gambar 4.9 Diagram alir perhitungan jarak piksel

4.2.6. Program Perhitungan jarak piksel menjadi jarak sebenarnya

Kode Program 6: Mengubah jarak piksel menjadi jarak sebenarnya

```
Jarak = 3.17 - (0.0766*pitago); //hasil dari  
linear regresi dan lack of fit  
Jarak1 = Jarak;  
cout << "jarak benda " << Jarak << endl;
```

Setelah diperoleh hasil jarak piksel maka dilakukan pengukuran terhadap instrument benda terhadap jarak, dengan menempatkan objek yang dijadikan target per 0.1 m, lalu setelah diperoleh jarak piksel per 0.1 m, maka data yang diperoleh dianalisis dengan menggunakan linear regresi dengan lack of fit dengan *confidence Interval* sebesar 95%, sehingga lack of fit dari model persamaan harus lebih besar dari 0.05. Namun apabila setelah diperoleh model persamaan yang memiliki untuk mendapatkan jarak sebenarnya, lalu model persamaan yang diperoleh dengan hasil linear regresi tersebut, dimasukan kedalam program lalu dari hasil program tersebut akan muncul jarak sebenarnya. Diagram alir perubahan jarak piksel menuju jarak sebenarnya bisa dilihat pada gambar 4.10.



Gambar 4.10. Diagram Alir dari Perhitungan Jarak piksel menjadi jarak sebenarnya

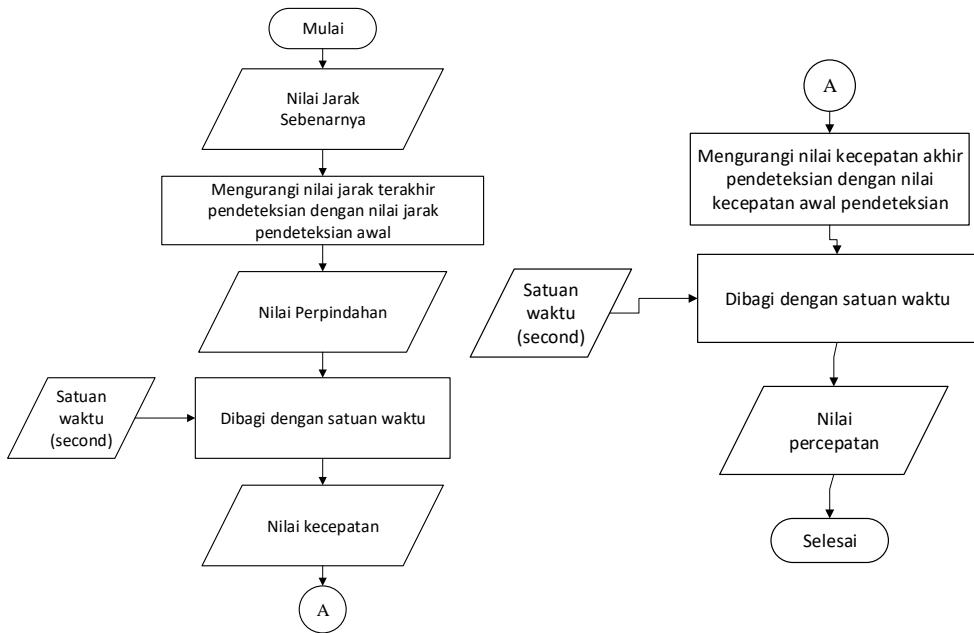
4.2.7. Program Perhitungan kecepatan dan percepatan

Kode Program 7: Perhitungan kecepatan dan percepatan

```
int Jarak0, Jarak1;

kec0 = 0;
clock_t t1, t2;
//t2 = clock();
printf("Time taken: %.2fs\n",
(double)(CLOCKS_PER_SEC *2) / CLOCKS_PER_SEC);
perpindahan = abs(Jarak1 - Jarak0)/100.0;
kecepatan = perpindahan / ((double)(clock() - t1)
/ CLOCKS_PER_SEC);
percepatan = (kecepatan - kec0) /
((double)(clock() - t1) / CLOCKS_PER_SEC);
Jarak0 = Jarak1;
kec0 = kecepatan;
cout << "perpindahan " << perpindahan << endl;
cout << "Kecepatan " << kecepatan << endl;
cout << "Percepatan " << percepatan << endl;
```

Setelah memperoleh hasil dari jarak sebenarnya maka, bisa dilanjutkan dengan perhitungan dalam program untuk memperoleh nilai kecepatan dan percepatan dengan membagi perpindahan terhadap waktu. Waktu yang digunakan adalah waktu komputer, lalu pada program digunakan perpindahan. Perpindahan diperoleh dengan mengurangi jarak akhir pendeksi dengan jarak awal pendeksi, lalu untuk mendapatkan kecepatan, nilai perpindahan yang diperoleh pada program dibagi dengan satuan waktu komputer yang dijalankan dalam program dan untuk memperoleh nilai percepatan, dengan menghitung kecepatan awal pada program dengan kecepatan akhir pada program yang terdeteksi. Kecepatan akhir dikurangi dengan kecepatan awal lalu dibagi dengan nilai waktu komputer yang terdeteksi

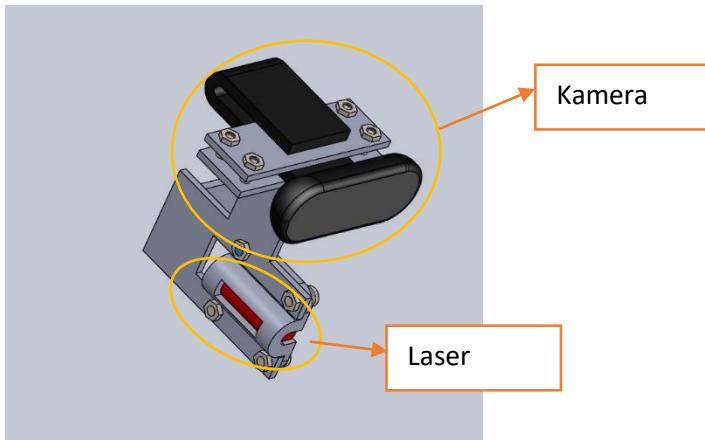


Gambar 4.11 Diagram alir dari perhitungan kecepatan dan percepatan

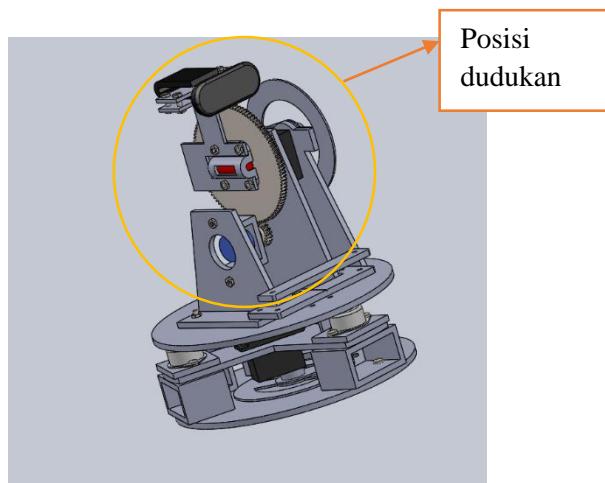
4.3. Konsep Instrumen Penelitian

Ide konsep perancangan instrumen berasal dari senjata militeristik yang umumnya menggunakan laser untuk menempatkan target, senjata militeristik tersebut ditambah sebuah kamera untuk mengukur pergerakan yang terdeteksi. Sehingga, dalam penelitian ini penulis merancang *prototype turret gun* kamera pada pendekatan dan laser. Kamera yang terhubung dengan laser diberi dudukan pada *prototype turret gun*. Sistem yang digunakan adalah pengukuran perpindahan posisi yang digunakan sebagai *feedback* agar mendapatkan nilai kecepatan, percepatan dan posisi dari objek. Sistem pendekatan telah didesain penulis seperti pada gambar 4.12. Desain pada gambar

4.12, terlihat kamera dan laser diberi dudukan sehingga posisi kamera dan laser tidak berpindah – pindah dan pada gambar 4.13 terlihat bahwa, dudukan laser dan kamera untuk pendeksiian, dipasangkan di gear elevasi sehingga dudukan antara laser dan kamera tersebut, bergerak menyesuaikan objek yang ditembakkan.



Gambar 4.12 Desain sistem pendeksiian posisi dari *prototype* *turret gun*



Gambar 4.13. Desain prototype *turrent gun*

BAB V

PENGUJIAN INSTRUMEN

5.1 Pendekstrian Objek

Dalam mendekstrian objek warna ganda yang di tempatkan pada target perlu dilakukan kalibrasi sehingga objek yang ingin didekstrian berbeda dengan objek lain yang tertangkap kamera. Pendekstrian objek yang berwarna merah cenderung memiliki warna cerah yang identik dengan background sehingga proses pendekstrian tidak bisa mendapatkan tepat pada pendekstrian objek. Sehingga saat pengujian dilakukan di terang hari maka perlu adanya pengaturan HSV dari kamera agar mendapatkan target yang jelas dalam kamera. Sehingga saat pendekstrian terlihat pada gambar 5.1.

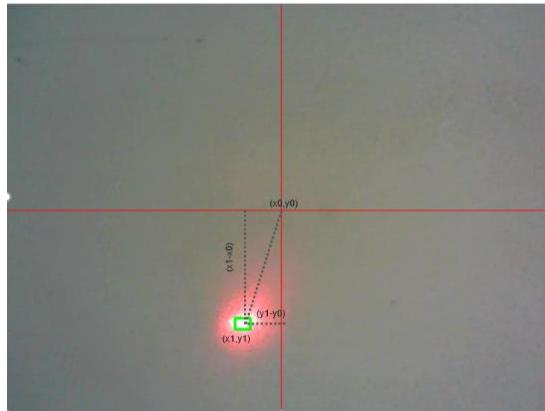


Gambar 5.1. Hasil Pendekstrian Objek.

5.2 Pengukuran Jarak

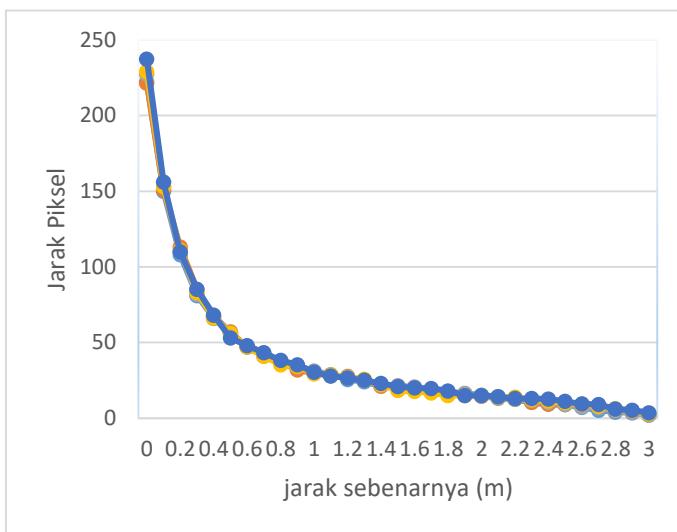
Pengujian program pengukuran perpindahan posisi instrumen dilakukan dengan meletakkan objek warna, dimana jarak tangkap objek menuju kamera akan divariasikan, sesuai dengan kebutuhan

dari pengambilan data. Data yang diperoleh adalah sebuah koordinat dari pendekatan objek, koordinat akan semakin menuju titik tengah (0,0) apabila jarak dari objek menjauhi kamera, untuk mendapatkan jarak piksel dari titik tengah maka perlu dihitung dengan teorema pythagoras dengan koordinat x dan y, sehingga, perhitungan teorema jarak x dan jarak y akan mendapatkan jarak piksel dari gambar yang ditangkap oleh kamera. Ilustrasi perhitungan terlihat pada gambar 5.2



Gambar 5.2. Ilustrasi perhitungan jarak piksel dalam gambar

Pengukuran jarak piksel dilakukan untuk mendapatkan model regresi agar mendapatkan jarak sebenarnya yang akan dimasukkan ke dalam program. Lalu dilakukan pengukuran untuk memperoleh jarak piksel per 0.1 m, sehingga setelah memperoleh data maka dibuat grafik yang terlihat pada gambar 5.3 dibawah ini, agar menentukan linear atau tidaknya data jarak piksel yang diperoleh. Pengukuran jarak piksel dilakukan sebanyak lima kali.



Gambar 5.3 Grafik jarak piksel per 0.1 m

Setelah diperoleh jarak piksel maka perlu dilakukan pengujian linear regresi terhadap data tersebut, sehingga dengan memasang $\alpha = 0.05$, maka linear regresi tersebut harus memiliki $\text{Lack of fit} > 0.05$, agar data jarak piksel tersebut gagal ditolak, apabila $\text{Lack of fit} < 0.05$ maka harus melakukan pemotongan data yang dianggap tidak linear hingga memperoleh data dengan lack of fit yang lebih besar dari 0.05. dari data jarak piksel yang telah diperoleh maka di analisis model linear regresinya dengan minitab. Dimana hipotesis adalah sebagai berikut:

- H_0 : model linear orde pertama tepat
- H_1 : model linear orde pertama tidak tepat
- $P\text{-value} < \alpha$, H_0 ditolak
- $P\text{-value} > \alpha$, H_0 gagal ditolak.

Regression Analysis: jarak real versus piksel

The regression equation is
 jarak real = 2.04 - 0.0142 piksel

Predictor	Coef	SE Coef	T	P
Constant	2.03998	0.06129	33.29	0.000
piksel	-0.014171	0.001008	-14.06	0.000

S = 0.594627 R-Sq = 56.4% R-Sq(adj) = 56.1%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	1	69.902	69.902	197.70	0.000
Residual Error	153	54.098	0.354		
Lack of Fit	140	54.038	0.386	83.63	0.000
Pure Error	13	0.060	0.005		
Total	154	124.000			

Dari hasil analisis minitab diatas ditemukan bahwa p-value lack of fit dari hasil keseluruhan data adalah 0.000 sehingga data pengukuran jarak piksel ditolak. Maka dari itu, perlu ada pemotongan data yang dianggap tidak linear, sehingga dilakukan analisis dengan minitab tetapi, data yang dianggap tidak linear sehingga berikut ini adalah hasil analisis model regresi yang ditampilkan oleh minitab dengan data range dari 1m hingga 3m. Adapun proses hasil regresi hingga mendapatkan data range 1m hingga 3 m dilampirkan pada Lampiran E

Regression Analysis: jarak asli versus jarak piksel

The regression equation is
 jarak asli = 3.18 - 0.0760 jarak piksel

Predictor	Coef	SE Coef	T	P
Constant	3.17505	0.02102	151.01	0.000
jarak piksel	-0.076039	0.001213	-62.70	0.000

S = 0.0976825 R-Sq = 97.4% R-Sq(adj) = 97.4%

Analysis of Variance

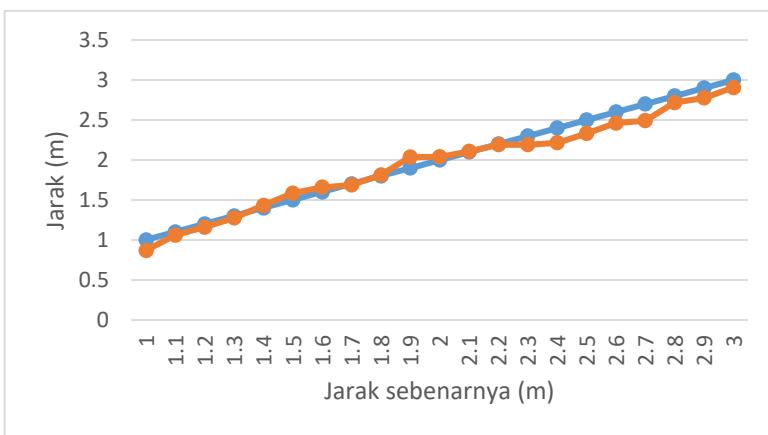
Source	DF	SS	MS	F	P
Regression	1	37.517	37.517	3931.85	0.000
Residual Error	103	0.983	0.010		
Lack of Fit	91	0.923	0.010	2.03	0.086
Pure Error	12	0.060	0.005		
Total	104	38.500			

Dan dari hasil analisis diatas maka diketahui lack of fit dari range 1m hingga 3 m adalah sebesar 0.086, dimana P-Value > 0.05, maka data jarak piksel antara 1m-3m merupakan data yang tidak mempunyai cukup bukti untuk ditolak atau gagal tolak sehingga model persamaan dari jarak asli = 3.18 - 0.0760 jarak piksel yang diperoleh dari hasil analisis regresi bisa digunakan dalam program, untuk mendapatkan jarak. Adapun hasil pengukuran jarak asli yang menggunakan model diatas diperoleh sebagai berikut.

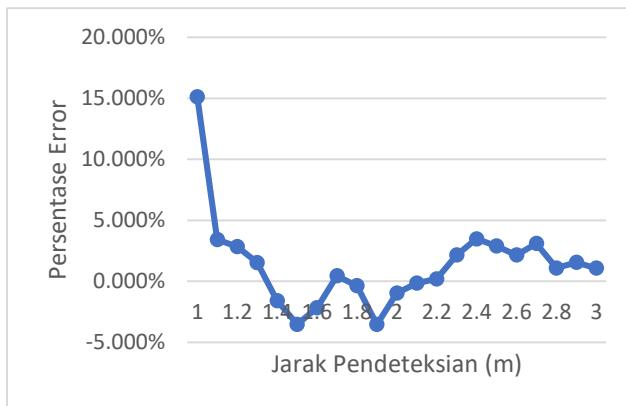
Tabel 5. 1 Tabel Pengukuran jarak asli dengan jarak dalam program dan Error

jarak asli	jarak dlm program	jarak piksel	error	Error %
1	0.8685968	30.4132	0.15128215992	15.128%
1.1	1.0600028	27.8947	0.03773310787	3.430%
1.2	1.1606268	26.5707	0.03392408309	2.827%
1.3	1.2745052	25.0723	0.02000368457	1.539%
1.4	1.432	23	-0.02234636872	-1.596%
1.5	1.584	21	-0.05303030303	-3.535%
1.6	1.658024	20.026	-0.03499587461	-2.187%
1.7	1.687208	19.642	0.00758175637	0.446%
1.8	1.812	18	-0.00662251656	-0.368%
1.9	2.037492	15.033	-0.06748100115	-3.552%
2	2.04	15	-0.01960784314	-0.980%
2.1	2.1070472	14.1178	-0.00334458573	-0.159%
2.2	2.1906244	13.0181	0.00427987564	0.195%
2.3	2.1914072	13.0078	0.04955391221	2.155%
2.4	2.2152484	12.6941	0.08339994738	3.475%
2.5	2.3315892	11.1633	0.07223004807	2.889%
2.6	2.46244144	9.44156	0.05586267262	2.149%
2.7	2.49181012	9.05513	0.08354965666	3.094%
2.8	2.71736824	6.08726	0.03040874578	1.086%
2.9	2.77533724	5.32451	0.04491805832	1.549%
3	2.90630804	3.60121	0.03223744996	1.075%

Dari tabel diatas terlihat hasil pengukuran dari jarak dalam program dengan jarak asli ditemukan error sehingga masih ditemukan kesalahan dalam pengukuran. Sehingga pengukuran jarak dengan menggunakan laser dan kamera masih belum terlalu akurat, lalu dengan memperhitungkan error yang ada maka diperoleh grafik error dengan grafik perbandingan jarak asli dengan jarak yang terdeteksi oleh program.



Gambar 5.4. Grafik perbandingan jarak asli dengan jarak yang terdeteksi oleh program



Gambar 5.5 Grafik Persentase error dari pengukuran jarak.

Terlihat pada dua grafik diatas terlihat bahwa error terdeteksi lebih dari 15% ketika di titik 1 m, lalu menurun hingga ke titik -5% di titik 1.5m lalu error cenderung meningkat dan fluktuatif hingga titik 2m yang memiliki error dibawah -1%, lalu dari titik 2 m error cenderung meningkat tajam hingga titik 2.4 m hingga ketitik error sebesar 9%, lalu cenderung fluktuatif dan menurun hingga titik 3m. Di titik 3m error sebesar 3%. Adapun rata-rata error sebesar 1.365%.

5.3. Pengukuran Kecepatan dan Percepatan

5.3.1 Pengukuran Kecepatan

Setelah melakukan pengukuran jarak maka dilanjutkan dengan pengukuran perpindahan posisi dengan mengukur kecepatan dan percepatan, Pengukuran kecepatan dan percepatan dilakukan dengan menggunakan *fletcher's trolley* sebagai alat verifikasi dari pengukuran kecepatan dan percepatan, sehingga hasil pendekesan percepatan dan kecepatan pada program yang terdeteksi dibandingkan dengan perhitungan teori dari perhitungan percepatan kecepatan dan percepatan dengan *fletcher's trolley*. *Fletcher's trolley* yang digunakan terbuat dari kayu dan pemberat

massa pertama terbuat dari kayu sehingga perlu memasukkan koefisien gesek antara kayu dengan kayu, dimana koefisien gesek (μ) antara kayu dengan kayu sebesar 0.4[13]. Setelah mendapatkan koefisien gesek :

$$\alpha = \frac{(m_2 - \mu m_1) g}{(m_1 + m_2)}$$

Dimana:

$$m_2 = 220 \text{ gram}$$

$$m_1 = 58 \text{ gr} + 170 \text{ gr} = 228 \text{ gram}$$

$$g = \text{gravitasi } (9.81 \frac{m}{s^2})$$

$$\alpha = \text{nilai percepatan}$$

$$\mu =$$

koefisien gesek kayu antar kayu (0.4)

Sehingga,

$$\alpha = \frac{(0.228 - 0.4 \times 0.220 \text{ kg}) \times 9.81 \frac{m}{s^2}}{0.228 + 0.220 \text{ kg}}$$

$$\alpha = 3.065625 \frac{m}{s^2}$$

Sehingga merujuk kepada titik kecepatan

$$v_x^2 = v_o^2 + 2\alpha s$$

Dimana:

$$s = \text{nilai perpindahan}$$

$$v_o = \text{kecepatan awal}$$

$$v_x = \text{kecepatan akhir}$$

Sehingga, diperoleh

$$v_x = \sqrt{0 + 2 \times 3.065625 \times (1.25041 - 1.18282)}$$

$$v_x = 0.848661602 \text{ m/s}$$

Maka diperoleh, tabel perhitungan teori adalah:

Tabel 5. 2 Tabel Hasil Perhitungan Teori dari Kecepatan dan Percepatan

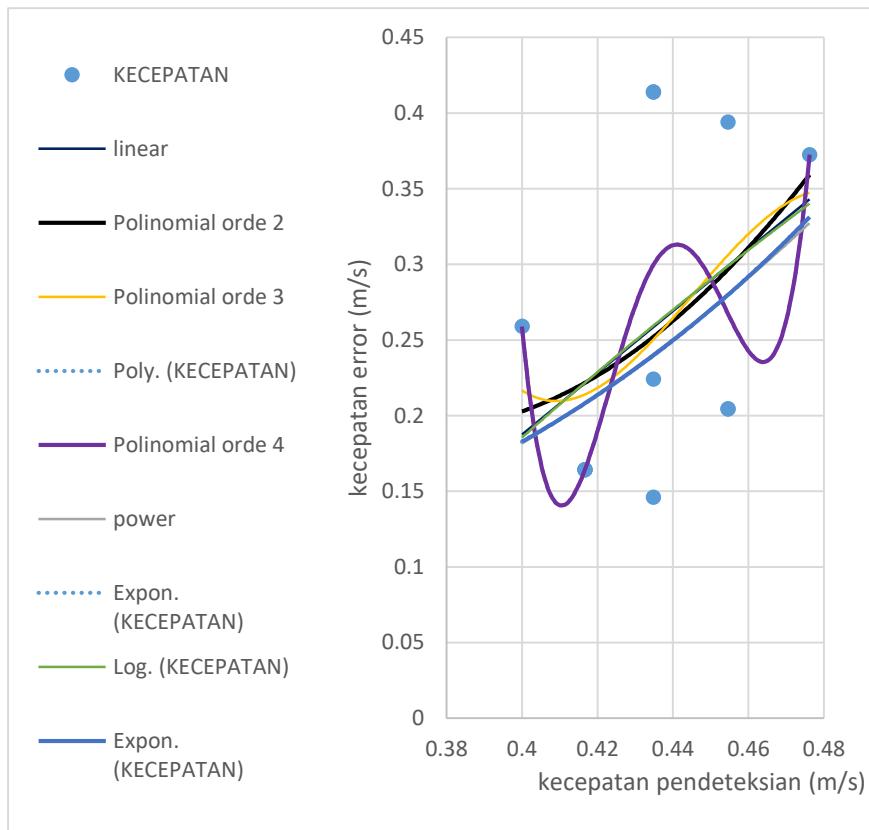
Teori		
Jarak (m)	Kecepatan (m/s)	Percepatan (m/s ²)
1.040274688	0.5808552	3.065625
1.104831563	0.6589768	3.065625
1.222187969	0.8486616	3.065625

Sehingga, dari tabel hasil perhitungan teori diatas lalu dibandingkan dengan hasil pendektsian dengan berupa grafik, dengan pengambilan data pendektsian dilakukan sebanyak 4 kali

Tabel 5. 3 Tabel Perbandingan Kecepatan pada Program dan Kecepatan Teori

Kecepatan			
Percobaan	Teori (m/s)	Pendeteksian (m/s)	Error
1	0.580855	0.416667	0.164188
	0.658977	0.4	0.258977
	0.848662	0.434783	0.413879
2	0.580855	0.434783	0.146072
	0.658977	0.454545	0.204432
	0.848662	0.47619	0.372472
3	0.580855	0.416667	0.164188
	0.658977	0.434783	0.224194
	0.848662	0.454545	0.394117
4	0.580855	0.416667	0.164188
	0.658977	0.454545	0.204432
	0.848662	0.434783	0.413879

Pada Tabel 5.3 terlihat bahwa perbandingan kecepatan pada program dengan kecepatan hasil perhitungan teori terlihat berbeda dikarenakan ditemukan error. Sehingga perlu mencari kompensasi error. Untuk menemukan error maka diperlukan perhitungan kompensasi error dengan memberikan regresi antara error dengan hasil pendektsian. Sehingga Perbedaan titik kecepatan terlihat pada gambar 5.6 dibawah



Gambar 5.6. Grafik Perbandingan Garis Regresi tiap titik pendekstrian kecepatan.

Terlihat pada grafik ada selisih antara regresi dengan titik kecepatan pendekstrian dengan error sehingga hasil regresi tersebut masih ditemukan error. Hasil analisis menunjukkan bahwa analisis regresi polynominal tingkat empat memiliki jumlah *sum of square* error yang paling sedikit sehingga metode regresi tersebut

paling tepat. Tabel error yang dihasilkan hasil regresi polynomial tingkat empat dapat dilihat pada tabel 5.4.:

Tabel 5. 4 Tabel Hasil Error kecepatan setelah dikompensasi dengan metode regresi

Kompesasi	SSE	Persentase
0.201728755	0.001409	6.071%
0.2944	0.001255	5.101%
0.339437322	0.005542	-9.615%
0.339437322	0.03739	24.975%
0.310295312	0.011207	13.841%
0.418166857	0.002088	5.109%
0.201728755	0.001409	6.071%
0.339437322	0.013281	14.885%
0.310295312	0.007026	-10.959%
0.201728755	0.001409	6.071%
0.310295312	0.011207	13.841%
0.339437322	0.005542	-9.615%

Adapun rata – rata error yang dihasilkan perhitungan kompensasi dengan menggunakan metode regresi polynomial tingkat 4 adalah sebesar 5.481%. Tabel perhitungan dengan metode regresi yang lain dilampirkan di halaman lampiran C

5.3.2. Pengukuran Percepatan

Setelah melakukan pengukuran kecepatan maka dilanjutkan dengan pengukuran perpindahan posisi dengan mengukur kecepatan dan percepatan. Pengukuran kecepatan dan percepatan dilakukan dengan menggunakan *fletcher's trolley* sebagai alat verifikasi dari pengukuran kecepatan dan percepatan, sehingga hasil pendekripsi percepatan dan kecepatan pada program yang terdeteksi dibandingkan dengan perhitungan teori dari perhitungan percepatan kecepatan dan percepatan dengan *fletcher's trolley*. *Fletcher's trolley* yang digunakan terbuat dari kayu dan pemberat massa pertama terbuat dari kayu sehingga perlu

memasukkan koefisien gesek antara kayu dengan kayu, dimana koefisien gesek (μ) antara kayu dengan kayu sebesar 0.4[13]. Adapun contoh perhitungan verifikasi dari percepatan adalah sebagai berikut:

$$\alpha = \frac{(m_2 - \mu m_1) g}{(m_1 + m_2)}$$

Karena mengubah sudut dari fletcher's trolley maka perlu mengubah perhitungan percepatan sehingga rumus percepatan mencari percepatan sudut menjadi:

$$\alpha = \frac{(m_2 - m_1 \sin \theta - \mu m_1 \cos \theta) g}{(m_1 + m_2)}$$

Dimana:

$m_1 = 220 \text{ gram}$

$m_2 = 58 \text{ gr} + 170 \text{ gr} = 228 \text{ gram}$

$g = \text{gravitasi } (9.81 \frac{m}{s^2})$

$\alpha = \text{nilai percepatan}$

$\mu = \text{koefisien gesek kayu antar kayu } (0.4)$

$\theta = \text{sudut yang terbentuk}$

Sehingga perhitungannya menjadi

$$\alpha = \frac{(228 - 220 \sin 0 - 0.4 \times 220 \cos 0) \times 9.81 \frac{m}{s^2}}{(220 \text{ gram} + 228 \text{ gram})}$$

$$\alpha = 3.065625 \frac{m}{s^2}$$

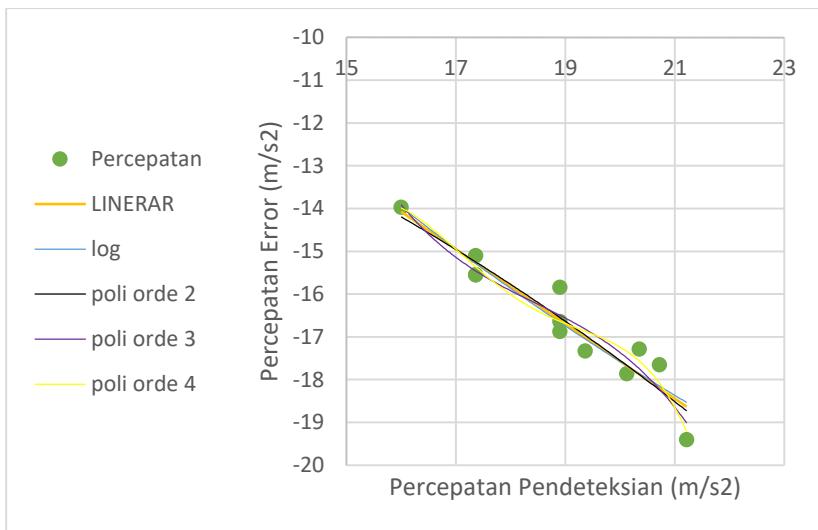
Setelah memperoleh besaran percepatan teori, maka melakukan pengujian pendektsian dari percepatan, karena prinsip dari fletcher trolley, nilai percepatan akan sama pada tiap titiknya sehingga tidak akan ada perubahan nilai percepatan hingga massa benda berhenti diujung katrol. Percobaan dilakukan sebanyak

empat kali di 4 sudut yang berbeda yakni $0^\circ, 10^\circ, 13^\circ, 16^\circ$. Adapun Hasil Pendektsian dan Perhitungan secara teori disajikan pada tabel dibawah

Tabel 5. 5Tabel Perhitungan Teori dan Pendektsian Percepatan

Percepatan			
sudut	Teori (m/s ²)	Pendektsian (m/s ²)	Error
0	3.065625	18.9036	-15.838
	3.065625	20.349256	-17.2836
	3.065625	20.72038	-17.6548
10	2.258365	17.3611	-15.1027
	2.258365	18.9036	-16.6452
	2.258365	20.123433	-17.8651
13	2.031331	16	-13.9687
	2.031331	18.9036	-16.8723
	2.031331	19.36241	-17.3311
16	1.812414	17.3611	-15.5487
	1.812414	17.3611	-15.5487
	1.812414	21.21755	-19.4051

Pada Tabel 5.5 terlihat bahwa perbandingan percepatan pada program dengan percepatan hasil perhitungan teori terlihat berbeda dikarenakan ditemukan error. Sehingga perlu mencari kompensasi error. Untuk menemukan error maka diperlukan perhitungan kompensasi error dengan memberikan regresi antara error dengan hasil pendektsian. Sehingga Perbedaan titik percepatan terlihat pada gambar 5.7 dibawah



Gambar 5.7. Grafik Perbandingan Garis Regresi tiap titik pendektsian percepatan

Terlihat pada grafik ada selisih antara regresi dengan titik kecepatan pendektsian dengan error sehingga hasil regresi tersebut masih ditemukan error. Hasil analisis menunjukkan bahwa analisis regresi polynomial tingkat dua memiliki jumlah *sum of square* error yang paling sedikit sehingga metode regresi tersebut paling tepat. Tabel error yang dihasilkan hasil regresi polynomial tingkat dua dapat dilihat pada tabel 5.6:

Tabel 5. 6 Tabel Hasil Error percepatan setelah dikompensasi dengan metode regresi

Kompesasi	SSE	Persentase
-16.550862	0.508207	-2.0107%
-17.880896	0.356726	-1.5623%
-18.239194	0.341569	-1.5001%
-15.246959	0.020801	-0.4423%
-16.550862	0.008906	0.2662%
-17.666247	0.039529	0.5261%
-14.1952	0.051316	-0.7502%
-16.550862	0.103303	0.9065%
-16.961657	0.136472	1.0170%
-15.246959	0.091039	0.9253%
-15.246959	0.091039	0.9253%
-18.729975	0.455843	1.6901%

Adapun rata – rata error yang dihasilkan perhitungan kompensasi dengan menggunakan metode regresi polynomial tingkat dua adalah sebesar -0.0008%. Tabel perhitungan dengan metode regresi yang lain dilampirkan di halaman lampiran D

(Halaman ini sengaja dikosongkan)

BAB VI

PENUTUP

6.1. Kesimpulan

Beberapa kesimpulan yang diperoleh pada penelitian tugas akhir ini adalah sebagai berikut:

1. Diperoleh cara dengan hasil pengukuran posisi dengan menggunakan laser dan mono-kamera, dengan rata – rata error yang dihasilkan adalah sebesar 1.365%.
2. Diperoleh cara hasil pengukuran kecepatan dengan menggunakan laser dan mono-kamera dengan rata – rata error yang dihasilkan adalah sebesar 5.481%.
3. Diperoleh cara hasil pengukuran percepatan dengan menggunakan laser dan mono-kamera dengan rata – rata error yang dihasilkan sebesar -0.0008%.

6.2. Saran dan Evaluasi

Berdasarkan hasil pengujian program dan pembahasan pada tugas akhir ini, muncul beberapa saran dan evaluasi untuk mengembangkan penelitian rancang bangun pengukuran perpindahan posisi, antara lain:

1. Perlu adanya kajian lagi mengenai penelitian pengukuran perpindahan posisi dengan menggunakan laser, dikarenakan warna intesitas laser akan terurai oleh cahaya terang sehingga, mempersulit untuk mendapatkan nilai HSV yang tepat dalam untuk mendapatkan titik cahaya
2. Perlu adanya kamera yang beresolusi tinggi dalam melaksanakan penelitian ini, karena semakin canggih kamera akan semakin mudah dalam menangkap laser yang terdeteksi oleh kamera
3. Dibutuhkan alat verifikasi kecepatan dan percepatan yang lebih tepat dibandingkan dengan penelitian *fletcher's trolley*.
4. Mencari metode pengukuran dengan kamera yang lebih akurat dibanding penggunaan laser

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] Pasa, F. P. 2017. **Penentuan Koordinat 3 Dimensi Target Tunggal Pada Sistem Pelontar Peluru Autotracking Menggunakan Stereo Kamera Bersudut Beta 15 Derajat.** Surabaya: Institut Teknologi Sepuluh Nopember.
- [2] Wiyagi, R. O. dan Mustar.Y.O. 2015. **Deteksi Jarak Objek Berbahaya Secara Real Time Menggunakan Kamera Tunggal.**Yogyakarta: Universitas Muhammadiyah Yogyakarta
- [3] Utama,S.N, Muslim M.A dan Setyawati. O. 2012. **Analisis Perhitungan Laser Rangefinder Menggunakan Persamaan Geometri Pada Sistem Keamanan Ruangan.** Malang: Universitas Brawijaya 1,1:150-154
- [4] Tsani, N. H.,, Buhanuddin D, Ir. dan Prasasti A.L., ST, MT.2017. **Implementasi Deteksi Kecepatan Kendaraan Menggunakan Kamera Webcam dengan Metode Frame Difference.** Bandung: Universitas Telkom 1,1:2373-2381
- [5] Anonim. 2017.**Kamera: Bagian & Fungsi, Prinsip Kerja, Pembentukan Bayangan, Contoh Soal dan Pembahasan.**
[<URL:https://www.fisikabc.com/2017/12/kamera.html>](https://www.fisikabc.com/2017/12/kamera.html)
- [6] Putra, D. 2010. **Pengolahan Citra Digital.** Yogyakarta: ANDI.
- [7] Gonzalez, R. C., dan Woods R. E. 2002. **Digital Image Processing.** New Jersey: Prentice-Hall.
- [8] Korn GA, Korn TM. 1961. **Mathematical Handbook for Scientists and Engineers.** New York: McGraw-Hill.
- [9] Basuki, A, dan Nana R. 2004. **Metode Numerik Sebagai Algoritma Komputasi, Diktat Kuliah D4, PENS- ITS.** Surabaya: PENS

- [10] Winahju, W. S. 2013. **Analisis Variansi dan Statistik Matematika Yang Terkait.** Surabaya: Departemen Statistika ITS
- [11] Abdullah, M. 2016. **Fisika Dasar I.** Bandung: Institut Teknologi Bandung
- [12] University of Calgary.2014. **Lesson - Fletcher's Trolley The Applet Simulates The Motion Of Two Blocks Connected By A String That Is Guided Over A Pulley.<<http://canu.ucalgary.ca/map/classes/info/forces/atwood/AtwooDApplet/fletcher/lesson.html>>**
- [13] Winingrah, P.H dan Hidayati.2017.**Eksperimen Gaya Gesek Untuk Menguji Nilai Koefisien Gesekan Statis Kayu Pada Kayu Dengan Program Matlab.** Yogyakarta: Universitas Sarjanawiyata Tamansiswa

LAMPIRAN A

Lampiran ini terdiri dari hasil pengukuran jarak piksel yang dilakukan agar mendapatkan model linear regresi untuk mencari jarak pada program

jarak asli	jarak piksel	jarak asli	jarak piksel
0	222.327	0	229.047
0.1	150.003	0.1	153.029
0.2	108.019	0.2	111.015
0.3	81.0542	0.3	83.006
0.4	66	0.4	66.0601
0.5	56.003	0.5	56.0803
0.6	47.0169	0.6	47.0925
0.7	42.2807	0.7	41
0.8	37.4115	0.8	35.2278
0.9	34.5524	0.9	34.2655
1	31.1141	1	29.42
1.1	28.799	1.1	28.6142
1.2	25.632	1.2	26.8739
1.3	24	1.3	25.7083
1.4	21.4803	1.4	22.2143
1.5	20.1359	1.5	18.26
1.6	19.43	1.6	17.72
1.7	18.4193	1.7	16.673
1.8	17.72	1.8	15.0333
1.9	16.2798	1.9	15
2	15.2552	2	14.9743
2.1	14.3178	2.1	14
2.2	12.5602	2.2	13.8843
2.3	10.44	2.3	12.3063
2.4	9.47	2.4	11.4018

2.5	9.04	2.5	10.3063
2.6	7.07	2.6	9.24453
2.7	5.32451	2.7	8.06324
2.8	4	2.8	6.34
2.9	3.60121	2.9	5.09873
3	2	3	3
0	221.329	0	237.221
0.1	151	0.1	156.021
0.2	113.018	0.2	110.018
0.3	85.02341	0.3	85.093
0.4	67.0761	0.4	68.061
0.5	57	0.5	53
0.6	47.0911	0.6	48.0097
0.7	41	0.7	43.2097
0.8	37.053	0.8	38.2275
0.9	32	0.9	35.3433
1	29.6142	1	30.4132
1.1	28.04423	1.1	27.8947
1.2	27.563	1.2	26.5707
1.3	25	1.3	25.0723
1.4	21.1131	1.4	23
1.5	20.8806	1.5	21
1.6	19.4132	1.6	20.026
1.7	17.72	1.7	19.642
1.8	16.72	1.8	18
1.9	15.2993	1.9	15.033
2	14.5602	2	15
2.1	13.991	2.1	14.1178
2.2	12.7043	2.2	13.0181
2.3	10.4403	2.3	13.0078

2.4	9.2134	2.4	12.6941
2.5	9.05	2.5	11.1633
2.6	8.26	2.6	9.44156
2.7	7.07	2.7	9.05513
2.8	5	2.8	6.08726
2.9	4	2.9	5.32451
3	3	3	3.60121
0	227.375		
0.1	153.012		
0.2	110.015		
0.3	82		
0.4	68		
0.5	56.1673		
0.6	47.1699		
0.7	42.107		
0.8	37.4715		
0.9	33.9056		
1	31.0046		
1.1	28.632		
1.2	27.3013		
1.3	25		
1.4	22.1359		
1.5	21.587		
1.6	20.587		
1.7	17.9747		
1.8	16.9234		
1.9	16.5242		
2	14.5602		
2.1	13		
2.2	12.3643		

2.3	11.707
2.4	10.443
2.5	9.08
2.6	7.548
2.7	7
2.8	5
2.9	4.1732
3	2

LAMPIRAN B

Konstruksi program

```
#include "stdafx.h"
#include "cv.h"
#include "highgui.h"
#include "opencv2\imgproc\imgproc_c.h"
#include <fstream>
#include "math.h"
#include <time.h>
#include <iostream>
#include <string>
#include <string.h>

using namespace std;

#define PI 3.14159265

int _tmain(int argc, _TCHAR* argv[])
{
    using namespace std;
    using namespace cv;
    ofstream myfile, myfile2, myfile3, myfile4;
    myfile.open("kecepatan.txt");
    myfile2.open("percepatan.txt");
    myfile3.open("detik.txt");
    myfile4.open("jarak.txt");
    Mat img, img_gray, img_HSV, img_gray2;
```

```
VideoCapture cam(0);
double distance = 0;
Point Posisi_laser; //Posisi objek
int jarakx =0 ,jaraky = 0, t_tengahx,t_tengahy;
//jarak objek dari titik tengah
double
theta,tan_theta,D,pitago,perpindahan,kecepatan,percepatan,J
arak,kec0,kec1;
int Jarak0, Jarak1;

FILE *data;
data = fopen("data320.csv","a");

cam.set(CV_CAP_PROP_FRAME_WIDTH, 640);
cam.set(CV_CAP_PROP_FRAME_HEIGHT, 480);
cam.set(CV_CAP_PROP_CONVERT_RGB, 1);
namedWindow("Frame", WINDOW_AUTOSIZE);
namedWindow("Biner", WINDOW_AUTOSIZE);

int iLowH = 0;
int iHighH = 360;

int iLowS = 0;
int iHighS = 255;

int iLowV = 223;
int iHighV = 255;

Jarak0 = 0;
kec0 = 0;
```

```
clock_t t1, t2;

while(waitKey(10) != 'a')
{
    //create a window called "Control"
    namedWindow("Control",
    CV_WINDOW_AUTOSIZE);

    /*Create trackbars in "Control" window*/
    cvCreateTrackbar("LowH", "Control",
    &iLowH, 360); //Hue (0 - 179) 26-125
    cvCreateTrackbar("HighH", "Control",
    &iHighH, 360);

    cvCreateTrackbar("LowS", "Control",
    &iLowS, 255); //Saturation (0 - 255) 83-255
    cvCreateTrackbar("HighS", "Control",
    &iHighS, 255);

    cvCreateTrackbar("LowV", "Control",
    &iLowV, 255); // Value(0 - 255) 85
    cvCreateTrackbar("HighV", "Control",
    &iHighV, 255);

    t1 = clock();
    cam >> img;
    cvtColor(img, img_gray,
    COLOR_BGR2GRAY);

    cvtColor(img, img_HSV,
    COLOR_BGR2HSV);
```

```
//Threshold the image
Mat imgThresholded;
Mat imgThresholded2;
inRange(img_HSV, Scalar(iLowH, iLowS,
iLowV), Scalar(iHighH, iHighS, iHighV),
imgThresholded);
inRange(img_HSV, Scalar(iLowH, iLowS,
iLowV), Scalar(iHighH, iHighS, iHighV),
imgThresholded2);

img_gray = imgThresholded;
img_gray2 = imgThresholded2;

erode(img_gray, img_gray, Mat(), Point(-
1,1), 4);
dilate(img_gray, img_gray, Mat(), Point(-
1,1), 4);
erode(img_gray2, img_gray2, Mat(), Point(-
1, 1), 4);
dilate(img_gray2, img_gray2, Mat(), Point(-
1, 1), 4);

vector<vector<Size>> contors;
vector<Vec4i> heirarcy;
findContours(img_gray, contors, heirarcy,
CV_RETR_EXTERNAL,
CV_CHAIN_APPROX_SIMPLE, Point(0,0));

vector<Rect> boundRect(contors.size());
vector<vector<Point>>
contor_poly(contors.size());
```

```
for(int i =0; i< contors.size(); i++)
{
    approxPolyDP(Mat(contors[i]),
contor_poly[i], 3, true);
    boundRect[i] =
boundingRect(Mat(contor_poly[i]));
}
int max_index = 0,max_area = 0;
for(int i =0; i< boundRect.size(); i++)
{
    int a = boundRect[i].area();
    //rectangle(img, boundRect[i].tl(),
boundRect[i].br(), Scalar(255,255,0), 2, 8, 0);
    if( a < max_area)
    {
        max_area = a;
        max_index = i;
    }
}
int confidence = 0;
for(int i=0; i< boundRect.size(); i++)
{
    if((boundRect[i].x <
boundRect[max_index].x + boundRect[max_index].width
&& boundRect[i].x > boundRect[max_index].x -
int(0.1*boundRect[max_index].width)) && (boundRect[i].y
> boundRect[max_index].y) )
        confidence += 45;
}
if (boundRect.size() > 0)
```

```
{  
    if (confidence > 99)  
        confidence = 0;  
    try{  
        Mat sub_image = Mat(img,  
Rect(max(boundRect[max_index].x - 30, 0),  
max(boundRect[max_index].y - 30, 0),  
min(int(boundRect[max_index].width*1.75), img.cols -  
boundRect[max_index].x + 30),  
min(boundRect[max_index].height * 3, img.rows -  
boundRect[max_index].y + 30));  
        //imshow("Frame",  
sub_image);  
    }  
    catch (int e){  
        cout << "Error occured" <<  
endl;  
    }  
    rectangle(img,  
boundRect[max_index].tl(), boundRect[max_index].br(),  
Scalar(255, 140, 0), 2, 8, 0);  
  
    fprintf(data, "%d , %d , %d\n",  
boundRect[max_index].width,  
boundRect[max_index].height,  
boundRect[max_index].area());  
    //Posisi_laser =  
    boundRect[max_index].tl() + (boundRect[max_index].br() -  
boundRect[max_index].tl());  
    t_tengahx = boundRect[max_index].x  
+ (boundRect[max_index].width / 2);
```

```

t_tengahy = boundRect[max_index].y
+ (boundRect[max_index].height / 2);
jaraky = t_tengahy - 240;
jarakx = t_tengahx - 320;
pitago = sqrt((jarakx * jarakx) +
(jaraky * jaraky));

Jarak = 3.17 - (0.0766*pitago); //hasil
dari linear regresi dan lack of fit
Jarak1 = Jarak;

//t2 = clock();
printf("Time taken: %.2fs\n",
(double)(CLOCKS_PER_SEC * 2) / CLOCKS_PER_SEC);
if (Jarak < 0)
{
    perpindahan = 0;
    kecepatan = 0;
    percepatan = 0;
}
else
{
    perpindahan = abs(Jarak1 -
Jarak0) / 100.0;
    kecepatan = ((perpindahan) /
((double)(clock() - t1) / CLOCKS_PER_SEC));
    percepatan = ((kecepatan -
kec0) / ((double)(clock() - t1) / CLOCKS_PER_SEC));
}

//theta = 0.0023*jarak + 0.0154;
//tan_theta = tan(theta * PI / 180.0);

```

```
//D = 0.07 / tan_theta;
//cout << "Posisi " << Posisi_laser <<
" Jarak selisih " << jarak << " Sudut " << theta << endl;
//cout << " Posisi Objek TL " <<
boundRect[max_index].tl() << " POsisi Objek BR " <<
boundRect[max_index].br() << endl;
//cout << "Posisi x objek " <<
t_tengahx << " Posisi y objek " << t_tengahy << endl;
//cout << "jarak x dari titik tengah "
<< jarakx << "jarak y dari titik tengah " << jaraky << endl;
cout << "jarak benda " << Jarak <<
endl;
//cout << "hasil pitagoras " << pitago
<< endl;
cout << "perpindahan " <<
perpindahan << endl;
cout << "Kecepatan " << kecepatan
<< endl;
cout << "Percepatan " << percepatan
<< endl;

line(img, Point(320, 0), Point(320,
480), Scalar(0, 0, 255), 1.5, 8);
line(img, Point(0, 240), Point(640,
240), Scalar(0, 0, 255), 1.5, 8);

std::stringstream ss1;
std::stringstream ss2;
std::stringstream ss3;
std::stringstream ss4;
std::stringstream ss5;
std::stringstream ss6;
```

```
if (Jarak < 0)
{
    ss1 << "jarak=" << "error";
    string disp1 = ss1.str();
    putText(img, disp1, Point(10,
420), FONT_HERSHEY_PLAIN, 0.85, CV_RGB(255, 255,
75), 1.5);
    ss5 << "Kecepatan=" <<
kecepatan;
    string disp5 = ss5.str();
    putText(img, disp5, Point(10,
435), FONT_HERSHEY_PLAIN, 0.85, CV_RGB(75, 255,
225), 1.5);
    ss6 << "Percepatan=" <<
percepatan;
    string disp6 = ss6.str();
    putText(img, disp6, Point(10,
450), FONT_HERSHEY_PLAIN, 0.85, CV_RGB(255, 255,
255), 1.5);
}
else
{
    //std::stringstream ss2;
    ss2 << "Jarak=" << Jarak;
    string disp2 = ss2.str();
    putText(img, disp2, Point(10,
420), FONT_HERSHEY_PLAIN, 0.85, CV_RGB(255, 255,
75), 1.5);

    //ss1 << "Jarak=" << Jarak;
    //string disp1 = ss1.str();
```

```
//putText(img, disp1, Point(10, 420),
FONT_HERSHEY_PLAIN, 0.85, CV_RGB(255, 255, 75),
1.5);
//std::stringstream ss3;
ss3 << "Kecepatan=" << kecepatan;
string disp3 = ss3.str();
putText(img, disp3, Point(10, 435),
FONT_HERSHEY_PLAIN, 0.85, CV_RGB(75, 255, 225),
1.5);
//std::stringstream ss4;
ss4 << "Percepatan=" << percepatan;
string disp4 = ss4.str();
putText(img, disp4, Point(10, 450),
FONT_HERSHEY_PLAIN, 0.85, CV_RGB(255, 255, 255),
1.5);
}
//distance =
1*pow(boundRect[max_index].area(), 0.468);

//cout << distance << " cm." <<
Confidence: "<<confidence<<endl;

Jarak0 = Jarak1;
kec0 = kecepatan;
imshow("Frame", img);
//imshow("Biner", img_gray);

myfile << kecepatan << "m/s" <<
endl;
myfile2 << percepatan << "m/s^2" <<
endl;
```

```
 myfile3 << t1 << "milliseconds" <<
endl;
 myfile4 << Jarak << "m" << endl;

}

else
    imshow("Frame", img);
//imshow("Biner", img_gray);
 imshow("HSV", img_HSV);
 imshow("biner non-edge detection",
img_gray2);

}

//fflush(data);
//fclose(data);
cam.release();
myfile.close();
myfile2.close();
myfile3.close();
myfile4.close();
return 0;
}
```

Lampiran Perhitungan Kecepatan

LAMPIRAN C

Percobaan	Teori	Kesepatan			linear			Polinomial tingkat 2		
		Pendeteksian	Error	Kompesasi	SSE	Percentase	Kompesasi	SSE	Percentase	
1	0.580555	0.4166667	0.164188	0.2214257	0.003276	8.97%	0.22169491	0.003307	9.01%	
	0.658977	0.4	0.258977	0.1874	0.005123	-12.19%	0.20278	0.003158	-9.32%	
	0.848662	0.434783	0.413879	0.2584095	0.024171	-22.43%	0.25198848	0.026208	-23.57%	
2	0.580555	0.434783	0.146072	0.2584095	0.01262	16.21%	0.25198848	0.011218	15.42%	
	0.658977	0.454545	0.204432	0.2987536	0.008897	12.52%	0.29559801	0.008495	12.27%	
	0.848662	0.47619	0.372472	0.3429419	0.008872	-3.61%	0.35530345	0.000173	-1.58%	
3	0.580555	0.4166667	0.164188	0.2214257	0.003276	8.97%	0.22169491	0.003307	9.01%	
	0.658977	0.434783	0.224194	0.2584095	0.01171	4.94%	0.25198848	0.000773	4.05%	
	0.848662	0.454545	0.394117	0.2987536	0.009094	-12.66%	0.29559801	0.00951	-12.98%	
4	0.580555	0.4166667	0.164188	0.2214257	0.003276	8.97%	0.22169491	0.003307	9.01%	
	0.658977	0.454545	0.204432	0.2987536	0.008897	12.52%	0.29559801	0.008495	12.27%	
	0.848662	0.434783	0.413879	0.2584095	0.024171	-22.43%	0.25198848	0.026208	-23.57%	
			Total	0.104843		total	0.104559			

Lampiran Perhitungan Kecepatan (Lanjutan Pertama)

Polinomial Tingkat 3				Polinomial Tingkat 4				POWER		
Kompesasi	SSE	Persentase	Kompesasi	SSE	Persentase	Kompesasi	SSE	Persentase		
0.216366108	0.002723	8.243%	0.201728755	0.001409	6.071%	0.208413119	0.001956	7.075%		
0.219	0.001598	-6.458%	0.2944	0.001255	5.101%	0.181555543	0.005994	-13.313%		
0.252671792	0.025988	-23.450%	0.339437322	0.005542	-9.615%	0.240652219	0.030007	-25.647%		
0.252671792	0.011363	15.506%	0.339437322	0.03739	24.975%	0.240652219	0.00845	14.003%		
0.308665183	0.010865	13.657%	0.310295312	0.011207	13.841%	0.279655934	0.005559	10.246%		
0.350355563	0.000494	-2.691%	0.418166857	0.002088	5.109%	0.32277	0.002043	-5.666%		
0.216366108	0.002723	8.243%	0.201728755	0.001409	6.071%	0.208413119	0.001956	7.075%		
0.252671792	0.000811	4.143%	0.339437322	0.013281	14.885%	0.240652219	0.000271	2.437%		
0.308665183	0.007302	-11.196%	0.310295312	0.007026	-10.959%	0.279655934	0.013101	-15.589%		
0.216366108	0.002723	8.243%	0.201728755	0.001409	6.071%	0.208413119	0.001956	7.075%		
0.308665183	0.010865	13.657%	0.310295312	0.011207	13.841%	0.279655934	0.005559	10.246%		
0.252671792	0.025988	-23.450%	0.339437322	0.005542	-9.615%	0.240652219	0.030007	-25.647%		
total	0.103441	total		0.098765	5.481% total		0.107555			

Lampiran Perhitungan Kecepatan (Lanjutan Kedua)

EKSPONENSIAL		logaritmik	
Kompesasi SSE	Persentase	Kompesasi SSE	Persentase
0.20684	0.001819	6.84%	0.221635
0.181606	0.005986	-13.30%	0.185433
0.238259	0.030842	-26.09%	0.259377
0.238259	0.008498	13.70%	0.259377
0.278	0.005412	10.04%	0.298795
0.329172	0.001875	-5.38%	0.340049
0.20684	0.001819	6.84%	0.221635
0.238259	0.000198	2.09%	0.259377
0.278	0.013483	-15.85%	0.298795
0.20684	0.001819	6.84%	0.221635
0.278	0.005412	10.04%	0.298795
0.238259	0.030842	-26.09%	0.259377
total	0.108007	total	0.105073

LAMPIRAN D

Lampiran Perhitungan Percepatan

sudut	Percepatan			Linear			Polynomial tingkat 2		
	Teori	Pendeteksian	Error	Kompesasi	SSE	Persentase	Kompesasi	SSE	Persentase
0	3.065625	18.9036	-15.838	-16.60809	0.593084	-2.17%	-16.550852	0.508207	-2.0107%
	3.065625	20.349256	-17.2836	-17.87088	0.3444856	-1.54%	-17.880896	0.356726	-1.5623%
	3.065625	20.72038	-17.6548	-18.19505	0.291921	-1.39%	-18.239194	0.341569	-1.5001%
10	2.258365	17.3611	-15.1027	-15.26072	0.02496	-0.48%	-15.246959	0.020801	-0.4423%
	2.258365	18.9036	-16.6452	-16.60809	0.001379	0.10%	-16.550852	0.008906	0.2662%
	2.258365	20.123433	-17.8651	-17.67362	0.036653	0.51%	-17.666247	0.039529	0.5261%
13	2.031331	16	-13.9687	-14.0718	0.010636	-0.34%	-14.1952	0.051316	-0.7502%
	2.031331	18.9036	-16.8723	-16.60809	0.069788	0.74%	-16.550852	0.103303	0.9065%
	2.031331	19.36241	-17.3311	-17.00887	0.103822	0.89%	-16.961657	0.136472	1.0170%
16	1.812414	17.3611	-15.5487	-15.26072	0.082924	0.88%	-15.246959	0.091039	0.9253%
	1.812414	17.3611	-15.5487	-15.26072	0.082924	0.88%	-15.246959	0.091039	0.9253%
	1.812414	21.21755	-19.4051	-18.62933	0.601875	1.95%	-18.729975	0.455843	1.6901%
				Total Error	2.244821		Total Error	2.204751	-0.0008%

Lampiran Perhitungan Percepatan (Lanjutan Pertama)

Polinomial tingkat 3			Polinomial tingkat 4		
Kompesasi	SSE	Persentase Kompesasi	SSE	Persentase	
-16.24869923	0.168694	-1.1684%	-20.68153395	23.46006	-12.2358%
-17.4303935	0.021539	-0.3885%	-22.98225752	32.47434	-13.1512%
-17.8904718	0.055562	-0.6105%	-23.90526666	39.0689	-14.0065%
-15.26796113	0.0273	-0.5064%	-18.24186506	9.85414	-8.8170%
-16.24869923	0.15724	1.1280%	-20.68153395	16.29171	-10.1965%
-17.19017896	0.455475	1.8087%	-22.52485167	21.71359	-10.9261%
-13.7644	0.041726	0.6863%	-16.0632	4.38706	-6.5325%
-16.24869923	0.388839	1.7739%	-20.68153395	14.5105	-9.6230%
-16.54991485	0.610217	2.1752%	-21.31555174	15.87602	-9.7952%
-15.26796113	0.078806	0.8604%	-18.24186506	7.253213	-7.5645%
-15.26796113	0.078806	0.8604%	-18.24186506	7.253213	-7.5645%
-18.6551908	0.557927	1.8732%	-25.61727658	38.59069	-13.2639%
Total Error	2.642134		Total Error	230.7334	

Lampiran Perhitungan Percepatan (Lanjutan Kedua)

logaritmik			Polinomial tingkat 5		
Kompesasi	SSE	Percentase	Kompesasi	SSE	Percentase
-16.6679021	0.688779	-2.333%	63.3069648	6263.921	-178.2409%
-17.8624489	0.33503	-1.515%	87.8804243	11059.48	-155.7267%
-18.1554188	0.250664	-1.288%	95.3904974	12779.23	-151.3929%
-15.2881019	0.034361	-0.568%	42.5180668	3320.157	-229.0451%
-16.6679021	0.000514	-0.064%	63.3069648	6392.354	-180.0589%
-17.6815552	0.033677	0.485%	83.3800012	10250.56	-160.0546%
-13.9646632	1.6E-05	0.013%	28.7056	1821.093	-335.8698%
-16.6679021	0.041766	0.575%	63.3069648	6428.71	-180.5702%
-17.0566337	0.075318	0.754%	69.9749283	7622.339	-172.4988%
-15.2881019	0.067904	0.798%	42.5180668	3371.748	-230.8178%
-15.2881019	0.067904	0.798%	42.5180668	3371.748	-230.8178%
-18.5397727	0.748854	2.177%	104.211498	15281.07	-148.9466%
total error	2.344787				

LAMPIRAN E**Lampiran Hasil Analisis Regresi****Regression Analysis: jarakasli (0.9) versus piksel(0.9)**

The regression equation is

$$\text{jarakasli (0.9)} = 3.13 - 0.0726 \text{ piksel(0.9)}$$

Predictor	Coef	SE Coef	T	P
Constant	3.13304	0.02313	135.47	0.000
piksel(0.9)	-0.072593	0.001255	-57.84	0.000

$$S = 0.113220 \quad R-\text{Sq} = 96.9\% \quad R-\text{Sq}(\text{adj}) = 96.8\%$$

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	1	42.891	42.891	3345.89	0.000
Residual Error	108	1.384	0.013		
Lack of Fit	96	1.324	0.014	2.76	0.027
Pure Error	12	0.060	0.005		
Total	109	44.275			

Regression Analysis: jarak asli (0.8) versus piksel (0.8)

The regression equation is

$$\text{jarak asli (0.8)} = 3.09 - 0.0691 \text{ piksel (0.8)}$$

Predictor	Coef	SE Coef	T	P
Constant	3.08798	0.02516	122.73	0.000
piksel (0.8)	-0.069067	0.001283	-53.83	0.000

$$S = 0.129630 \quad R-Sq = 96.2\% \quad R-Sq(\text{adj}) = 96.2\%$$

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	1	48.701	48.701	2898.20	0.000
Residual Error	113	1.899	0.017		
Lack of Fit	101	1.839	0.018	3.64	0.008
Pure Error	12	0.060	0.005		
Total	114	50.600			

Regression Analysis: jarak asli (0.7) versus piksel (0.7)

The regression equation is

$$\text{jarak asli (0.7)} = 3.03 - 0.0645 \text{ piksel (0.7)}$$

Predictor	Coef	SE Coef	T	P
Constant	3.02542	0.02861	105.76	0.000
piksel (0.7)	-0.064476	0.001361	-47.38	0.000

$$S = 0.156001 \quad R-Sq = 95.0\% \quad R-Sq(\text{adj}) = 95.0\%$$

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	1	54.628	54.628	2244.74	0.000
Residual Error	118	2.872	0.024		
Lack of Fit	105	2.812	0.027	5.80	0.001
Pure Error	13	0.060	0.005		
Total	119	57.500			

Regression Analysis: jarak asli (0.6) versus piksel (0.6)

The regression equation is

$$\text{jarak asli (0.6)} = 2.95 - 0.0594 \text{ piksel (0.6)}$$

Predictor	Coef	SE Coef	T	P
Constant	2.95234	0.03231	91.39	0.000
piksel (0.6)	-0.059423	0.001426	-41.68	0.000

$$S = 0.186910 \quad R-\text{Sq} = 93.4\% \quad R-\text{Sq}(\text{adj}) = 93.3\%$$

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	1	60.703	60.703	1737.57	0.000
Residual Error	123	4.297	0.035		
Lack of Fit	110	4.237	0.039	8.35	0.000
Pure Error	13	0.060	0.005		
Total	124	65.000			

BIODATA PENULIS



Penulis yang bernama Ikhsan Abi Nubli biasa dipanggil Abi dilahirkan di Jakarta pada tanggal 24 Desember 1995 sebagai anak pertama dari tiga bersaudara dari pasangan Fiqih Syamsu Rokhman dan Elmasa Wahyuni. Penulis bertempat tinggal di Jalan Galaxi 3 no 333, RT 06/RW 010, Jakasetia, Bekasi Selatan, Bekasi. Penulis telah menempuh pendidikan formal dimulai dari SDI Darussalam (2001-2007), Global Prestasi JHS (2007-2010), SMA Labschool Jakarta (2010-2013), dan pada tahun 2013 penulis melanjutkan studi di Departemen Teknik Mesin ITS melalui jalur SBMPTN. Selama perkuliahan, penulis aktif mengikuti kegiatan di KM ITS. Penulis pernah bergabung dalam organisasi kemahasiswaan yaitu sebagai Staff Departemen Kewirausahaan HMM-ITS periode 2014/2015, dan Kepala Biro Usaha Internal HMM-ITS periode 2016/2017. Selain itu penulis juga aktif mengikuti kepanitiaan yaitu diantaranya sebagai IC Gerakan Integralistik (GERIGI) ITS tahun 2015 dan menjadi ketua OC *Advanced Leadership Training* (ALT) 2016. Penulis mendapatkan kesempatan Kerja Praktik di PT Dirgantara Indonesia pada tahun 2017. Penulis mempu Penulis menerima segala saran dan kritik dari segala pihak yang bersifat membangun dan dapat meningkatkan manfaat Tugas Akhir ini melalui email penulis abi.nubli17@gmail.com / abi.nubli12@hotmail.com